

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M2612 – Elektrotechnika a informatika
Studijní obor: 3902T005 – Automatické řízení a inženýrská
informatika

Automatická identifikace drážních vagónů a nákladních vozů
(Automatic trail wagons and trucks identification)

Diplomová práce

Autor: Jan Matějčíček
Vedoucí práce: Ing. Lukáš Matela, Ph.D.

Rozsah práce a příloh:

Počet stran: 83
Počet obrázků: 52
Počet vzorců: 19
Počet tabulek: 5

V Liberci 15. května 2008

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Ústav řízení systémů a spolehlivosti

Akademický rok: 2007/08

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení: **Jan Matějčíček**

studijní program: M 2612 – Elektrotechnika a informatika

obor: 3902T005 - Automatické řízení a inženýrská informatika

Vedoucí ústavu Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu:

Automatická identifikace drážních vagónů a nákladních vozů

Zásady pro vypracování:

1. Seznamte se s knihovnami NI-IMAQdx a NI-VISION a jejich použitím v prostředí LabWindows/CVI
2. Seznamte se s transformacemi barevných prostorů pro získání optimálně kontrastních snímků
3. Seznamte se s využitím skupiny funkcí souvisejících s technikou OCR a specifikujte limity využití těchto funkcí
4. Seznamte se možnostmi synchronizace digitálních kamer s rozhraním IEEE 1394 a GigE
5. Realizujte aplikaci pro spolehlivou identifikaci drážních vagónů a nákladních vozů a zpracujte dokumentaci v potřebném rozsahu

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

- [1] Hlaváč, V., Sedláček, M.: Zpracování signálů a obrazů, ČVUT FEL Praha 2000
- [2] Fischer, J.: Optoelektronické senzory a videometrie, ČVUT FEL Praha 2002
- [3] Schlesinger, Hlaváč: Deset přednášek z teorie statistického a strukturního rozpoznávání, ČVUT FEL Praha 1999

Vedoucí diplomové práce:

Ing. Lukáš Matela, Ph.D.

Konzultant:

Ing. František Kratochvíl

Zadání diplomové práce:

15.10.2007

Termín odevzdání diplomové práce:

16. 5. 2008



Doc. Ing. Libor Tůma, CSc.

.....

vedoucí ústavu



Prof. Dr. Ing. Jirí Maryška, CSc.

.....

děkan

V Liberci dne 25.10.2007

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím své diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne

Podpis

PODĚKOVÁNÍ:

Na tomto místě bych velice rád poděkoval svému vedoucímu práce Ing. Lukáši Matelovi, Ph.D. za trpělivé a odborné vedení, cenné rady a připomínky k práci.

Dále bych chtěl poděkovat své rodině a přítelkyni za vytvoření dobrého studijního zázemí a poskytnutí podpory během celého studia.

ANOTACE:

Cílem této diplomové práce bylo vypracovat algoritmy pro lokalizaci a identifikaci textu na drážních nákladních vagónech určených pro převoz dřeva. Další částí diplomové práce bylo rozeznat a identifikovat registrační značky nákladních automobilů pro firmu NEZÁVISLÁ PŘEJÍMKA s.r.o., pobočka Štětí. To vše ve vývojovém prostředí LabWindows/CVI, ve kterém jsou již implementovány základní i složitější funkce pro zpracování obrazu. Hlavní knihovnou pro práci s obrazem, včetně OCR, je knihovna NIVISION, jejíž použité funkce jsou v této diplomové práci také popsány.

V teoretické části diplomové práce jsou zpracovány základní informace o zpracování obrazu a použité postupy. Je zde popsán celý systém včetně použitých kamer, senzorů a programu pro automatickou identifikaci. Také je proveden popis principů jednotlivých identifikačních modulů včetně vysvětlení řešení jednotlivých dílčích problémů.

ANOTATION:

This diploma thesis is aimed to work up algorithms for localization and identification of text occurring on wagons destined for wood transport. Subject matter of another part of this diploma thesis is recognition and identification of registration numbers of trucks, ordered by the company NEZÁVISLÁ PŘEJÍMKA s.r.o., branch office in Štětí. All these works were based upon the LabWindows/CVI development environment, where basic as well as more complicated functions for image processing are implemented there already. The NIVISION library was used as the main library for image processing inclusive OCR. Its capacities applied are described in this diploma thesis as well.

In theoretical part of this diploma thesis, basic information on image processing as well as procedures of work used are worked up there. The whole system inclusive the cameras, sensors and software for automatic identification applied is described here. Demonstration of principles for the individual identification modules and explanation of solutions of individual particular problems is included as well.

OBSAH:

Úvod	- 9 -
1 Zpracování obrazu	- 10 -
1.1 Digitální interpretace obrazu	- 10 -
1.2 Digitalizace	- 10 -
1.2.1 Vzorkování	- 10 -
1.2.2 Kvantování.....	- 11 -
1.3 Barevné prostory	- 12 -
1.3.1 RGB	- 13 -
1.3.2 Grayscale – Y model	- 14 -
1.3.3 HSL.....	- 15 -
1.4 Jasové transformace a prahování obrazu	- 15 -
1.4.1 Histogram	- 16 -
1.4.2 Šedotónové jasové transformace	- 16 -
1.4.3 Prahování podle histogramu	- 17 -
1.4.4 Prahování obrazu metodou NIBlack	- 19 -
1.5 Geometrické transformace obrazu	- 19 -
1.5.1 Transformace souřadnic	- 20 -
1.5.2 Jasové aproximace.....	- 21 -
1.6 Binární morfologie obrazu	- 25 -
1.6.1 Eroze.....	- 25 -
1.6.2 Dilatace	- 25 -
1.6.3 Otevření	- 26 -
1.6.4 Uzavření.....	- 26 -
2 Kameraný systém	- 27 -
2.1 Připojení kamer	- 27 -
2.2 Použité kamery	- 29 -
2.3 Objektivy	- 31 -
2.4 Popis orientace použitých kamer	- 32 -
2.5 Senzory	- 34 -
2.6 Použité PC	- 35 -
2.7 Softwarová část	- 37 -
2.7.1 Identifikace kolejí	- 37 -
2.7.2 Verifikace ID vagónu	- 37 -
2.7.3 Automatické přijímání emailů	- 38 -
3 Knihovna NI – VISION.....	- 39 -
3.1 Důležité funkce	- 39 -
3.2 OCR Training.....	- 40 -

3.2.1	Znaková sada	- 40 -
3.2.2	Programový způsob vytváření znakové sady	- 40 -
3.2.3	Vytváření abecedy pomocí NI OCR Training Interface	- 41 -
3.2.4	Použití OCR.....	- 46 -
3.3	Vision Assistant 8.0	- 47 -
4	Modul čtení drážních vagónů	- 49 -
4.1	Omezení identifikace	- 49 -
4.2	Popis modulu.....	- 50 -
4.2.1	Rozlišení modrých vagónů	- 50 -
4.2.2	Problém lokalizace textu	- 51 -
4.2.2.1	1. způsob lokalizace – univerzálnější způsob.....	- 51 -
4.2.2.2	2. způsob lokalizace.....	- 53 -
4.2.3	Segmentace textu	- 58 -
4.2.4	Čtení textu.....	- 59 -
4.3	Struktura modulu.....	- 60 -
5	Modul čtení RZ nákladních automobilů	- 62 -
5.1	Umístění kamery	- 62 -
5.2	Lokalizace registrační značky.....	- 64 -
5.3	Transformace RZ	- 65 -
5.4	Čtení RZ.....	- 66 -
5.5	Popis funkce najdiRZ(...)	- 66 -
6	Závěr	- 67 -
	Použité zdroje	- 69 -

Slovník zkratek a symbolů:

AEC – Auto Exposure Control, tedy automatické řízení doby závěrky kamery.

AGC – Auto Gain Control, tedy automatické řízení zesílení kamery.

CCD – Označení typu snímače obrazové informace, z anglického Charge-Coupled Device (česky zařízení s vázanými náboji)

ČD – Označovací zkratka Českých Drah

DB – Označovací zkratka Deutsche Bahn, tedy německých Drah

DMA – Direct Memory Access – možnost přímého přístupu do paměti bez účasti procesoru

DPI – Dot Per Inch, údaj udávající počet obrazových bodů na palec, tedy rozlišení.

fps – frames per second, tedy frekvence snímkování

GigE – Gigabit Ethernet, tedy ethernetu s rychlostí přenosu 1 Gbit/s

HSL – Barevný model Hue–Saturation– Lightness (česky barevnost–saturace–světelnost)

ID – Označení číselného identifikátoru vagónu

IEEE – Institute of Electrical and Electronics Engineers (česky „Institut pro elektrotechnické a elektronické inženýrství“), správa standardů v elektrotechnice

IP – Stupeň krytí proti vniku cizích těles a vody

lx – jednotka osvětlení Lux, udává osvětlení světelným tokem na metr čtvereční
(1 lx = 1 lm/m² = 1 cd·sr·m⁻²).

NI – National Instruments, firma z USA.

NIBlack – Metoda lokálního tresholdu patentovaná NI.

OCR – Optical Character Recognition, tedy optické rozpoznávání znaků. Metoda umožňující automatické rozpoznávání a digitalizaci textu z obrazové předlohy.

OCV – Optical character verification, tedy možnost ověření kvality vytištěných znaků.

PoE – Power over Ethernet, napájení zařízení přes ethernet

RGB – Barevný model Red–Green–Blue (česky červená-zelená-modrá).

RTE –RunTime Engine, licence (pro provoz zkompileovaných programů z LabWindows a LabView)

RZ – Zkratka registrační značka automobilu.

USB – Universal Serial Bus, tedy universální sériová sběrnice.

UTP – Označení kabelu nestíněné kroucené dvoulinky

V – jednotka elektrického napětí (či elektrického potenciálu), $V = m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$

W – jednotka výkonu, $W = kg \cdot m^3 \cdot s^{-3}$

XGA – Extended Graphics Array, standard display firmy IBM (ekvivalent 1024x768)

ŽSR – Označovací zkratka Železnic Slovenskej republiky, tedy slovenských státních drah

Úvod

Použití kamerových systémů je v dnešní době jednou z nejrychleji se rozvíjejících oblastí průmyslové automatizace. Jedná se například o systémy pro automatickou kontrolu kvality výrobků, rozlišování jednotlivých druhů součástí nebo např. řízení a přesné navádění robotů. Jednou z dalších možností použití kamerových systémů je využití OCR, tedy automatické identifikace textu v obraze.

Právě problematikou OCR se zabývá tato diplomová práce, kdy se provádí automatická identifikace drážních vagónů a nákladních automobilů. Celý systém vznikl pro firmu NEZÁVISLÁ PŘEJÍMKA s.r.o., pobočka Štětí. Diplomová práce je zaměřena na práci s obrazem a čtení textu z obrazu. Přes její hlavní orientaci zde bude nastíněn i zbytek systému od hardwarových komponent až k ostatním částem celkového programu.

Úvodem této diplomové práce bych chtěl čtenáře provést základy zpracování obrazu. Jedná v první řadě o pořízení obrazu již v digitální podobě, popř. jeho převedení do digitální podoby – digitalizace. Dále bych vysvětlil základy barevné reprezentace obrazu a základní použité metody pro jeho zpracování.

Čtenář práce se také dozví o použitých kamerách, jejich možnostech komunikace a způsobech jejich řízení. V práci se také nachází popis počítače, pomocí kterého je celá aplikace realizovaná. Nalezne zde i popis celého systému, včetně detailního rozboru způsobu lokalizace textu a charakteristiky funkce OCR.

V práci jsou rozebrány dva způsoby použití OCR. Je zde popsán postup automatické lokalizace textu a čtení identifikačních znaků z nákladních drážních vagónů. Druhou možností použití OCR, který je uveden v této diplomové práci, je lokalizace a rozlišování registračních značek nákladních automobilů. Čtenář bude seznámen s postupy a způsoby řešení obou základních modelů automatické identifikace.

Kromě celého systému je zde i popis průběhu a výsledky nasazení do praxe, statistické výsledky a jejich vývoj. V textu práce se vyskytují názvy jednotlivých druhů drážních vagónů. Jsou součástí terminologie přijaté od zadavatelské firmy a neodpovídají tedy oficiálním názvům těchto vozů.

1 ZPRACOVÁNÍ OBRAZU

1.1 DIGITÁLNÍ INTERPRETACE OBRAZU

Člověk zpracovává obraz již od pradávna. V dnešní době se ale nedíváme na obraz jen jako na pěknou podívanou, ale chápeme ho jako vícerozměrnou spojitou skalární funkci f . Mluvíme o ní jako o obrazové funkci [3]. Tato funkce je dvourozměrná $f(x, y)$, pro statický záznam obrazu v rovině, popř. trojrozměrná $f(x, y, t)$, což je plošný obraz měnící se v čase nebo prostorový obraz $f(x, y, z)$.

Pokud obrazová funkce f má jednu návratovou hodnotu, hovoříme o monochromatickém obraze. Pokud je návratová hodnota tvořena vektorem, jedná se o obraz barevný. Barvou objektu nazýváme jeho schopnost odrážet elektromagnetické záření jen určité vlnové délky. Na kvalitě pořízeného obrazu se podílí více různých faktorů. Jedním z hlavních předpokladů pořízení kvalitního obrazu je znalost vlastností snímaného objektu, jako odrazivost a ozáření povrchu.

Obraz nemusí být vytvářen pouze z jasových hodnot fotoaparátu, kamery či scanneru, ale je možné získat obrazové informace i z jiné veličiny, jako je intenzita rentgenového záření, ultrazvuk nebo tepelné záření [8]. Pro lepší přehlednost budeme brát jako zdroj obrazu digitální kameru.

1.2 DIGITALIZACE

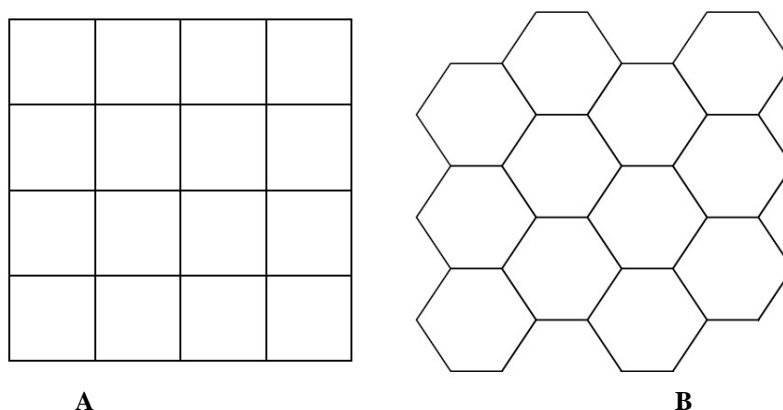
Protože je obraz reprezentován jako spojitá funkce, nelze ji v tomto tvaru počítačově zpracovat. Je tedy nutné převést obraz do digitální podoby. Tento převod se provádí ve dvou krocích. Nejprve je nutné provést vzorkování [1] na matici $M \times N$ bodů, a poté se provádí druhý krok digitalizace, kvantování [1] na K úrovní.

1.2.1 VZORKOVÁNÍ

Vzorkování vychází z potřeby převedení spojité obrazové funkce na diskrétní funkci o přesně daném rozměru. Vzorkovací krok je vzdálenost mezi dvěma nejbližšími sousedními vzorkovacími body. Tato vzdálenost musí vyhovovat Shannonově větě [2] o vzorkování, která říká, že vzorkovací frekvence musí být alespoň dvakrát vyšší, než je nejvyšší frekvence ve vzorkovaném signálu. Pro obraz tedy platí, že vzdálenost mezi těmito sousedními body

musí být minimálně dvakrát větší, než je nejmenší detail, který je pro nás v obraze důležitý. Pro nejlepší výsledky je optimální, pokud nejmenší relevantní objekt v obraze reprezentován větším množstvím pixelů. Rozlišení je však třeba volit s rozvahou, protože s vyšším rozlišení stoupá výpočetní náročnost zpracování obrazu [8] .

Velikost obrazu je poté udávána v obrazových bodech, tzv. pixelech a rozlišení obrazu je udáváno v počtu bodů na palec, tedy DPI (dot per inch). Tyto pixely jsou přesně geometricky určeny vzorkovací mřížkou. Jsou dva základní druhy mřížky, a to čtvercová a hexagonální viz. obr. 1.1.



obr. 1.1 – Vzorkovací mřížky: A – čtvercová mřížka, B – hexagonální mřížka

Použitím hexagonální mřížky je možné odstranění některých neduhů čtvercové mřížky, jako možnost provést přímku skrze uzavřenou křivku bez toho, aniž by přímka křivku protla [1]. Je zde ještě problém s měřením vzdálenosti v obraze. Nicméně snazší implementace výpočetních algoritmů přímo předurčuje čtvercovou mřížku pro častější použití při výrobě obrazových senzorů. Přes náročnější výpočty jsou snímače s hexagonálním uspořádáním vyráběny (např. firmou FUJI).

1.2.2 KVANTOVÁNÍ

Kvantováním přiřazujeme hodnotám jasu obrazové funkce odpovídající jasové úrovni [2]. Počet těchto jasových úrovní volíme podle potřeby nebo možností systému. Platí však podmínka, že jejich minimální počet je 50, protože při tomto a nižším počtu pro lidské oko vznikají tzv. falešné obrysy [8].

Většinou se pro určení počtu těchto úrovní k vychází z počtu bitů používaných pro tento převod b . Počet jasových úrovní se pak určí:

$$k = 2^b \quad (1.1)$$

Nejčastěji se používají hodnoty pro jeden, dva nebo osm bitů. Toto odpovídá dvěma, čtyřem a 256-ti jasovým úrovním. Často se u černobílých digitálních kamer vyskytuje i rozlišení 10 nebo 12 bitů.

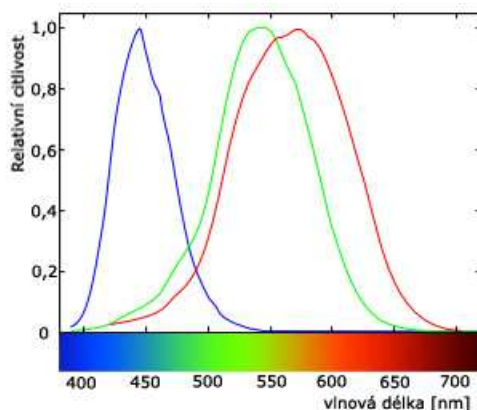
1.3 BAREVNÉ PROSTORY

Lidské oko svým vývojem dosáhlo toho, že je citlivé jen na velice úzkou část spektra elektromagnetického záření, kterému říkáme viditelné světlo. Viditelné světlo je v rozmezí vlnových délek od 380 nm (vidíme jako fialovou barvu) až do 700nm (barva červená). Toto viditelné spektrum se také nazývá optická škála [6] a je zobrazeno na obr. 1.2. Pokud je vlnová délka nižší než 380 nm hovoříme pak o ultrafialovém záření, pokud je vyšší než 700 nm, hovoříme o infračerveném záření.



obr. 1.2 – optická škála

Oko vnímá okolní barevný svět pomocí tří druhů barvocitlivých čípků. Kromě čípků jsou na sítnici lidského oka umístěny tzv. tyčinky. Ty jsou na barvy necitlivé, tedy barvoslepé [7]. Uplatňují se hlavně pro vidění ve tmě a vnímají světlo v modro–zelené světelné oblasti. Jednotlivé druhy barvocitlivých čípků jsou citlivé vždy jen na jeden druh barvy. Jsou to červená ($\lambda_R = 560$ nm), zelená ($\lambda_G = 530$ nm) a modrá ($\lambda_B = 440$ nm), z čehož vyplývá, že člověk vidí pouze třibarevně. Červené čípky jsou nejcitlivější na červeno–žlutou barvu, zelené na zeleno–žlutou a modré jsou nejcitlivější na modro–fialovou, jak je názorně vidět na obr. 1.3. Barvy, které vidíme, vznikají tak, že barvocitlivé čípky určí intenzitu jednotlivých základních barev a z nich je poté složena výsledná barva.



obr. 1.3 – citlivost lidského oka na barvy (zdroj: http://en.wikipedia.org/wiki/Image:Cones_SMJ2_E.svg)

Barva	Rozsah vlnových délek
Červená	~ 625 – 740 nm
Oranžová	~ 590 – 625 nm
Žlutá	~ 565 – 590 nm
Zelená	~ 500 – 565 nm
Azurová (cyan)	~ 485 – 500 nm
Modrá	~ 440 – 485 nm
Fialová	~ 380 – 440 nm

tab. 1.1 – vlnové délky viditelného světla

1.3.1 RGB

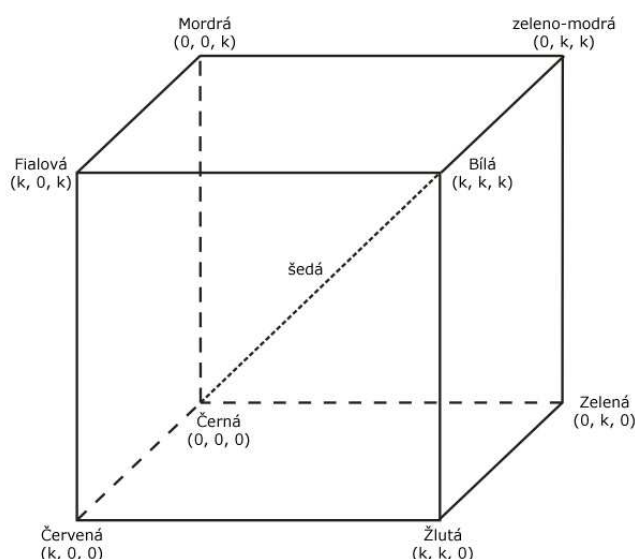
Základním barevným prostorem je prostor RGB, vycházející z lidského způsobu rozpoznávání barev. Z tohoto barevného modelu jsou následně odvozeny ostatní barevné modely. Na tento způsob pořízení barevného obrazu se konstruuje i digitální snímáče. Jsou zaznamenány tři základní barvy:

R (Red) – červená

G (Green) – zelená

B (Blue) – modrá

Kombinací těchto barev vznikají všechny ostatní barvy. Každý barevný pixel je tedy reprezentován vektorem $[R \ G \ B]$, který udává barevné složky.



obr. 1.3 – Prostor RGB

Barevný prostor RGB je teoreticky reprezentován barevnou krychlí na jejíž vrcholech jsou jednotlivé základní barvy, jak je patrné na obrázku 1.3. Maximální hodnoty složek

barevného vektoru jsou dány vzorcem 1.2, tedy pro osmibitovou reprezentaci je maximum 255.

$$k = 2^b - 1 \quad (1.2)$$

RGB model je možné rozšířit o složku A (alfa kanál), který nabývá hodnoty v intervalu (0, 1). Tato složka udává průhlednost barvy, kdy pro hodnotu 1 je barva plně průhledná.

1.3.2 GRAYSCALE – Y MODEL

Grayscale je šedotónový barevný prostor, kde je každý pixel reprezentován pouze jednou barevnou složkou. Tato složka určuje jas pixelu. Základní převodní vztah mezi RGB modelem a Y modelem je:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (1.3)$$

Označení Y vychází z definice trichromatické soustavy, kdy odpovídá hodnotě jasové funkce obrazu. Převod na Y model je zohledněn pro vidění barev lidským okem, protože oko je nejcitlivější na zelenou barvu a tudíž tvoří zelená složka téměř 60% šedotónového modelu. Největší nevýhodou tohoto převodu je fakt, že je nevratný (nemá inverzní transformaci). Počet jasových hodnot se určí jako u RGB modelu podle vzorce 1.2.



obr. 1.4 – rozdělení jasů v Y modelu pro 256 úrovní

Pro tuto diplomovou práci byla vytvořena jiná varianta Y modelu, parametrizovaná na získání nejlépe kontrastního snímku.

$$Y_R = 0,4 \cdot R + 0,6 \cdot G \quad (1.4)$$

Tento Y_R model je použit pro získání šedotónového obrazu pro identifikaci vagónů s červenou základovou barvou. Použití pouze červené složky se neosvědčilo kvůli vysokým hodnotám šumu. Díky konstrukci čipu použitých kamer se šum odstraní přidáním zelené složky, která má dvojnásobnou citlivost.

$$Y_B = 0,5 \cdot G + 0,5 \cdot B \quad (1.5)$$

Pro identifikaci modrých vagónů je použit Y_B model, určený podle vzorce 1.5. Vychází pouze z modré a zelené složky obrazu, protože červená má na těchto vozech minimální kontrast. Podrobněji dále.

1.3.3 HSL

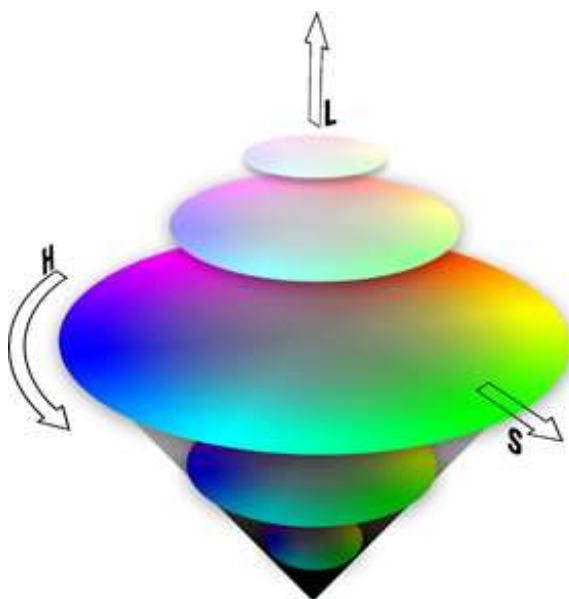
Dalším, v práci použitým barevným modelem je HSL model. Barva je zde interpretována opět třísložkovým barevným vektorem. Jsou to složky:

H (Hue) – barevnost

S (Saturation) – saturace

L (Lightness) – světllost

Barevnost udává barevný tón, tedy převládající barva udávaná jako úhel v intervalu $(0, 360)^\circ$. Saturace udává sytost barvy v rozsahu 0 (bílá) až 1 (odpovídající spektrální barva) a světllost udává množství světla v barvě v rozmezí 0 (barva ve tmě) až 1 (za plného světla). Vše je pro lepší názornost zobrazeno na obr. 1.5.



obr. 1.5 – model HSL (zdroj: http://upload.wikimedia.org/wikipedia/nl/4/4f/Kleurenkegel_HSL.png)

1.4 JASOVÉ TRANSFORMACE A PRAHOVÁNÍ OBRAZU

Jasové transformace jsou jednou ze základních metod předzpracování obrazu. Nejlepší způsob předzpracování je žádné předzpracování [2], tj. pořídit takový snímek, který již nevyžaduje žádné další úpravy. Tomuto ovšem často u automatizovaných systémů nelze

vyhovět, protože to například neumožňují různé světelné podmínky vnějšího prostředí během dne.

Díky jasovým transformacím už bohužel žádné informace z obrazu nezískáme, pouze můžeme zvýraznit ty, které jsou pro naše řešení důležité.

1.4.1 HISTOGRAM

Histogram [1] je ve většině případů zobrazován jako sloupcový graf. Vzniká tak, že se projde obraz pixel za pixel a podle hodnoty jasu každého z nich se zvýší hodnota v histogramu odpovídající tomuto jasu. Po prozkoumání celého obrazu tak vznikne graf, jenž má na x-ové ose hodnoty jasu (nejčastěji 0..255) a y-ové ose je počet obrazových bodů s daným jasnem (obr. 1.9).

Z histogramu obraz rekonstruovat nelze, protože neuchovává informace o poloze jednotlivých jasových hodnot, pouze jejich počet. Z toho vyplývá, že pro více různých obrazů je možné získat stejný histogram.

Je také možné vytvořit histogram pro barevný obraz, kdy se řeší každá barevná složka zvlášť. Vznikají tak histogramy pro každou barvu. Další možností je například průměrovat hodnoty barev a z nich poté získat histogram.

1.4.2 ŠEDOTÓNOVÉ JASOVÉ TRANSFORMACE

Šedotónové transformace [1] nezávisí na pozici pixelu v obraze. Provádí se transformace τ jasu původního obrazu p se souřadnicemi $[p_0, p_K]$ na novou hodnotu jasu q pro bod $[q_0, q_K]$.

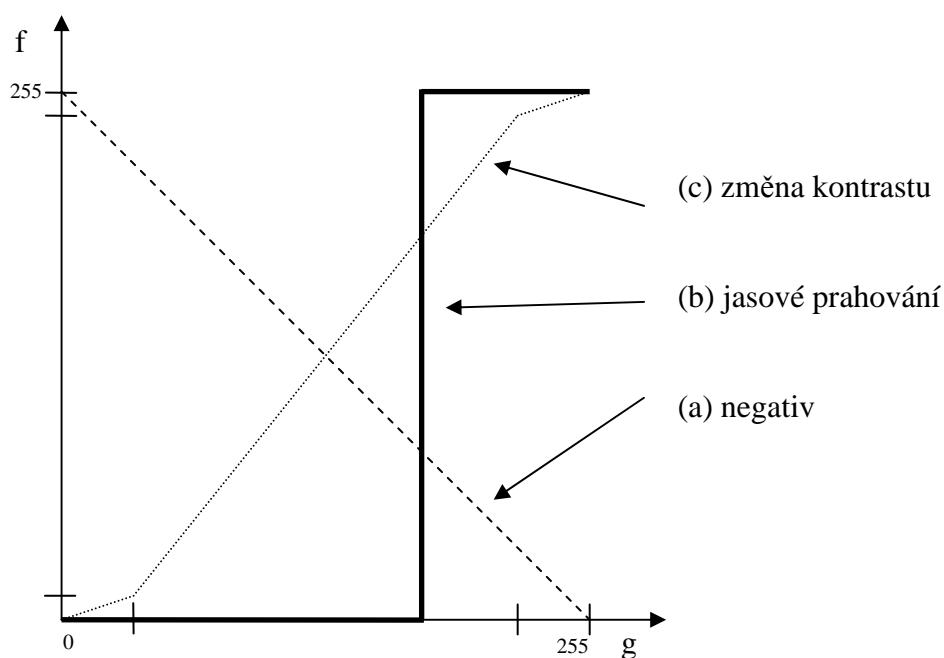
$$q = \tau(p) \quad (1.6)$$

Mezi základní šedotónové transformace patří negativ (obr. 1.6, křivka a). Tato transformace provede záměnu jasu ze světlých na tmavé hodnoty dle 1.7. Patří mezi vratné jasové transformace, je tedy možné získat původní obraz opětovným provedením negativu.

$$q = 255 - p \quad (1.7)$$

Další elementární transformací je jasové prahování (obr. 1.6, křivka b). V tomto případě má výsledný obraz pouze dvě hodnoty jasu (0 a 255, popř. 1), je to tedy převod na binární obraz. Pokud je hodnota jasu pixelu větší než prahová hodnota, je vyhodnocen jako

světlý, v opačném případě je vyhodnocen jako tmavý. Jasové prahování je ale nevratné, tzn. že zde dochází k trvalé ztrátě obrazových informací.



obr. 1.6 – základní šedotónové transformace

Poslední zde uvedenou šedotónovou transformací je změna kontrastu. Jejím prostřednictvím lze zvětšit kontrast obrazu, původně nekонтastního.

Převodové rovnice:

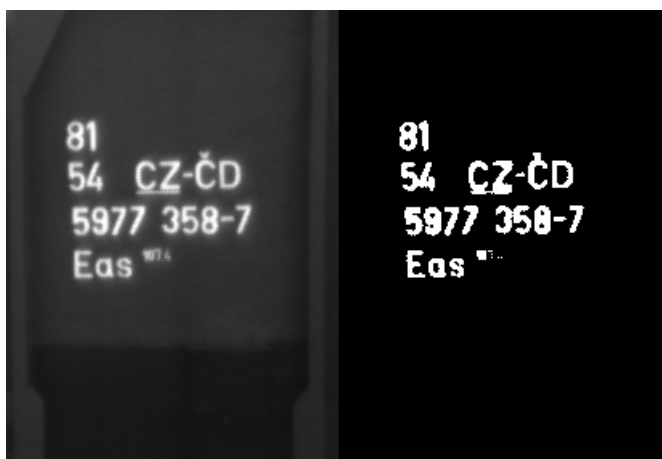
$$\begin{aligned}
 q(0, q_1) &= \frac{q_1}{p_1} \cdot p(0, p_1), \\
 q(q_1, q_2) &= \frac{q_2 - q_1}{p_2 - p_1} \cdot p(p_1, p_2), \\
 q(q_2, 255) &= \frac{255 - q_2}{255 - p_2} \cdot p(p_2, 255),
 \end{aligned}
 \tag{1.8}$$

pro jasový rozsah $(0, 255)$, q je výstupní funkce a p je vstupní jasová funkce. q_1 , q_2 , p_1 a p_2 jsou převodní koeficienty. Před použitím této transformace je nutné brát v úvahu možné splnutí blízkých hodnot jasu, a tedy k možné trvalé ztrátě obrazových informací.

1.4.3 PRAHOVÁNÍ PODLE HISTOGRAMU

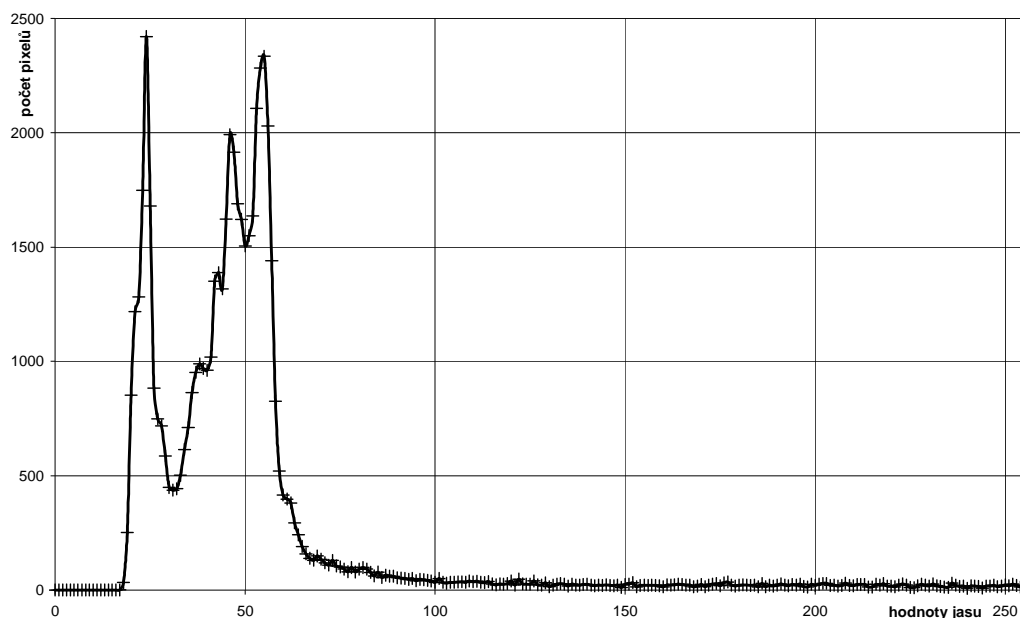
Prahování podle histogramu lze nazvat také jako procentní prahování. Při tomto prahování se vychází z histogramu. Určí se procentuální práh a podle tohoto prahu se zjistí

počet příznivých pixelů. Postupně sčítáme hodnoty z histogramu od nejvyššího nebo od nejnižšího jasu, podle požadavku na nalezení světlých nebo tmavých bodů. Tento součet poté porovnáváme s počtem příznivých pixelů. Pokud součet překročí mezní hodnotu, provedeme jasové prahování s hodnotou jasu, která odpovídá naposledy sčítanému jasu.



obr. 1.7 – odprahovaný obraz (7% práh)

Výsledkem je opět binární obraz, který nebyl získán pevně danou prahovou hodnotou. Prahovací hodnota je určena ze vstupního obrazu tak, abychom vždy získali přibližně stejný počet světlých bodů, bez ohledu na kontrast vstupního obrazu.



obr. 1.8 – histogram k obr. 1.7

1.4.4 PRAHOVÁNÍ OBRAZU METODOU NIBLACK

Metoda prahování obrazu NIBlack provádí lokální prahování. Nepracuje tedy s celým obrazem, ale vychází vždy jen z malého okolí prahovaného bodu. Je to adaptivní prahovací algoritmus, nejvhodnější pro prahování textu pro OCR nebo pro zpracování mapových podkladů.

Určení hodnoty lokálního prahu T je použit vzorec 1.9, kde m je lokální střední hodnota v bodě (x, y) a σ je lokální směrodatná odchylka vypočtená z lokálního okolí bodu (x, y) .

$$T(x, y) = m(x, y) + k \cdot \sigma(x, y) \quad (1.9)$$

Uživatelé definovaný parametr k udává stupeň výchyly. Nastavení tohoto parametru bývá zásadní pro kvalitu prahování. Zároveň je ho třeba zohlednit pro různé typy vyhodnocovaných snímků.



obr. 1.8 – Použití metody NIBlack s okolím 32x32 a hodnotou $k = 0,20$

1.5 GEOMETRICKÉ TRANSFORMACE OBRAZU

Běžnou součástí počítačové grafiky jsou geometrické transformace [2] a jejich použití v obrazové analýze je také samozřejmostí. Jejich pomocí lze opravit poškozený pořízený obraz nebo simulovat jiný úhel pohledu. Toto poškození nevzniká špatnými podmínkami, ale většinou díky jiným omezením. Jedná se například o optické zkreslení obrazu, způsobené pozicí kamery. Lze opravit např. zkosení nebo rotaci obrazu a v případě potřeby lze realizovat i takové věci, jako eliminaci zkreslení použité optiky.

Geometrické transformace převádí souřadnice bodu ve vstupním obraze na nové souřadnice ve výstupním obraze. Protože po převedení výstupních na vstupní souřadnice

nejdou většinou celé číslo, ale reálné, je obtížné určit přesnou hodnotu jas. Je proto nutné provést aproximaci této hodnoty. Geometrické transformace jsou tedy rozděleny na dvě části. Transformace souřadnic a aproximace jasové hodnoty.

Transformaci souřadnic lze vyjádřit obecným zápisem:

$$x' = \sum_{r=0}^m \sum_{k=0}^{m-r} a_{rk} x^r y^k \quad y' = \sum_{r=0}^m \sum_{k=0}^{m-r} b_{rk} x^r y^k. \quad (1.10)$$

Koeficienty a_{rk} , b_{rk} jsou parametry transformace, které lze určit metodou nejmenších čtverců při známé poloze dvojic bodů ve vstupním a výstupním obraze. Rovnice 1.10 je často nahrazována bilineární transformací. Pro jednodušší operace jako změna měřítka, zkosení a rotace se používá afinní zobrazení [1], které lze vyjádřit:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ b_1 & b_2 & b_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (1.11)$$

kde x', y' jsou množiny bodů ve výstupním obraze a x, y jsou množiny bodů ve vstupním obraze.

1.5.1 TRANSFORMACE SOUŘADNIC

Změna měřítka

Nezákladnější geometrickou transformací je změna měřítka. Touto operací lze změnit velikost obrazu, jeho výšku i šířku. Změna měřítka je popsána maticemi:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad (1.12)$$

kde x', y' jsou množiny bodů ve výstupním obraze a x, y jsou množiny bodů ve vstupním obraze. Koeficienty a, b udávají změnu měřítka osy x a y .

Zkosení

Zkosením se opravuje deformace zkoseného obrazu (podle osy x). Hodnoty osy y se v tomto případě nemění. Matice transformace je:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \tan(\varphi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\tan(\varphi) & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad (1.13)$$

kde x', y' jsou množiny bodů ve výstupním obraze a x, y jsou množiny bodů ve vstupním obraze. Úhel φ reprezentuje úhel zkosení.

Rotace

Rotace je geometrická transformace, při které se celý obraz pootočí o požadovaný úhel. Matice transformace je:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad (1.14)$$

kde x', y' jsou množiny bodů ve výstupním obraze a x, y jsou množiny bodů ve vstupním obraze. Úhel φ reprezentuje úhel rotace.

1.5.2 JASOVÉ APROXIMACE

Interpolace metodou nejbližšího souseda

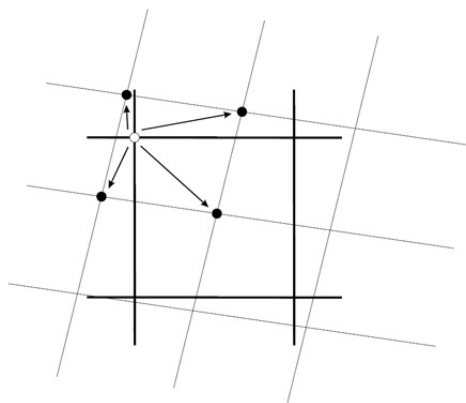
Při převodu z výstupního na vstupní obraz, hodnoty souřadnic bodu (x, y) nenabývají celočíselných hodnot. Není tedy jednoznačně určena hodnota jasu pro bod ve výstupním obraze. Hodnota jasu pro výstupní bod (x', y') obrazu je určena aproximací jasu v okolí bodu (x, y) ve vstupním obraze.

Interpolace metodou nejbližšího souseda je základní, jednoduchá a nejrychlejší metoda aproximace hodnot jasové funkce. Bodu (x, y) je přiřazena hodnota jasu odpovídající jasu nejbližšího bodu v celočíselné mřížce.

Tato interpolace lze popsat funkcí:

$$f_1(x, y) = g_s(\text{round}(x), \text{round}(y),) \quad (1.15)$$

kde f_1 udává jas a g je celočíselná mřížka. Hodnoty x, y jsou reálné souřadnice obrazu. Při použití této aproximace může vzniknout chyba, která má velikost maximálně půl pixelu (obr. 1.9).



obr. 1.9 – interpolace metodou nejbližšího souseda

Tato chyba je na obraze zřetelná u objektů s přímočarými obrysy natočenými šikmo vůči rastru [1].

Lineární interpolace

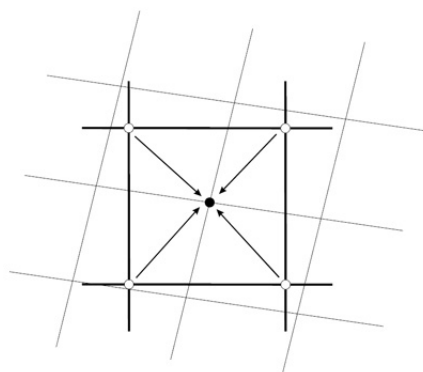
Lineární interpolace využívá pro určení hodnoty jasu čtyř bodů sousedících se zpracovávaným bodem (x, y) [1]. Vliv každého bodu je úměrný jeho vzdálenosti k vyhodnocovanému pixelu. K určení jeho jasu je použita jejich lineární kombinace, jak je vidět z obrázku obr. 1.10, vlevo. Lineární interpolace je dána vztahem:

$$f(x, y) = (1-a)(1-b)g(l, k) + a(1-b)g(l+1, k) + b(1-a)g(l, k+1) + abg(l+1, k+1),$$

$$\text{kde} \quad l = \text{round}(x), \quad a = x - l,$$

$$k = \text{round}(y), \quad b = y - k$$
(1.16)

f je vypočtený jas bodu (x, y) a g označuje celočíselnou mřížku, x, y jsou reálné souřadnice a l, k jsou souřadnice interpolačního jádra. Koeficienty a, b vyjadřují vzdálenost k bodům interpolačního jádra. Tato aproximace jasové hodnoty způsobuje jen malé snížení rozlišení, ale výstupní obraz je lehce rozmazán. To je způsobeno tím, že hodnoty obrazu jsou vyhlazovány lineárním filtrem. Přesto je lineární interpolace mnohem lepší než metoda nejbližšího souseda, protože zde není ona nepříjemná schodovitost. Nejvíce je tento rozdíl patrný na obrazech se šikmými čarami.



obr. 1.10 – lineární interpolace

Bilineární interpolace

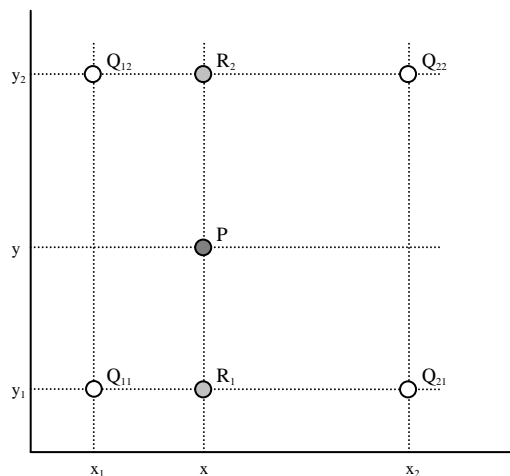
Bilineární interpolace [9] je rozšíření lineární interpolace jako interpolace ve dvou proměnných. Celé rozšíření spočívá v tom, že se provede lineární interpolace nejprve v jednom směru, a poté v druhém směru ovšem již s posunutým souřadným systémem. Nejprve se provádí interpolace ve směru osy x , tedy:

$$\begin{aligned}
 f(R_1) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad \text{kde } R_1 = (x, y_1), \\
 f(R_2) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad \text{kde } R_2 = (x, y_2), \\
 f(P) &\approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).
 \end{aligned} \tag{1.17}$$

Nyní lze určit požadovaný odhad:

$$\begin{aligned}
 f(x, y) &= \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \\
 &+ \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) + \\
 &+ \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \\
 &+ \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1).
 \end{aligned} \tag{1.18}$$

kde f je vypočtený jas bodu (x, y) a x, y jsou reálné souřadnice. Q_{11} , Q_{12} , Q_{21} a Q_{22} jsou body o známém jas, tedy okolí vypočítávaného bodu. Bilineární interpolace má lepší výsledky než lineární, ale za cenu větší časové náročnosti výpočtu.



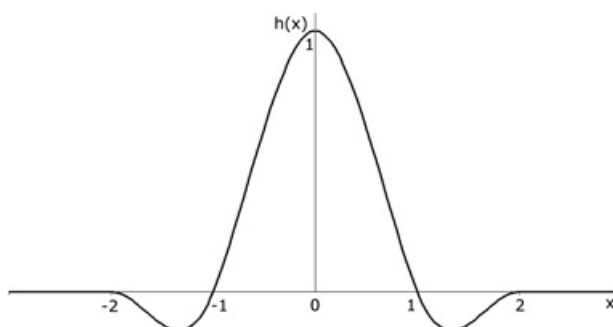
obr. 1.11 – bilineární interpolace

Bikubická interpolace

Bikubická interpolace [1] je o poznání složitější, než jasové aproximace předešlé. Rozšiřuje model obrazové funkce lokální interpolací bikubickým polynomem. Výpočet používá okolí šestnácti bodů. Pro složité dvojrozměrné vyjádření tohoto okolí je na obrázku obr. 1.12 pouze jednorozměrný případ. Výpočet je (1.19) uveden také jen jednorozměrně. Pro druhý rozměr by byl stejný.

$$h(x) = \begin{cases} 1 - 2 \cdot |x|^2 + |x|^3 & \text{pro } 0 \leq |x| < 1 \\ 4 - 8 \cdot |x| + 5 \cdot |x|^2 - |x|^3 & \text{pro } 1 \leq |x| < 2 \\ 0 & \text{jinde} \end{cases} \quad (1.19)$$

Výsledný jas je součtem součinů vypočtených $h(x)$ a $h(y)$ s hodnotou jasu odpovídajících bodů bikubického jádra.



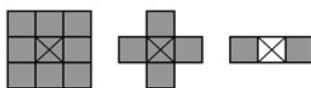
obr. 1.12 – jádro bikubické interpolace

Časová náročnost výpočtu je však vykoupena vysokou výslednou kvalitou obrazu, protože nejlépe aproximuje jas detailů. Tyto však musí být reprezentovány v původním obraze alespoň dvěma pixely.

1.6 BINÁRNÍ MORFOLOGIE OBRAZU

Obraz jako takový lze chápat jako bodovou množinu a tím vzniká možnost použít na obraz množinové funkce. Toto je základ matematické morfologie. Binární matematická morfologie [2] má situaci ještě usnadněnou díky práci pouze s binárním obrazem. Tento obsahuje pouze dvě barvy (černou (0) a bílou (1)) a je chápán jako 2D množina. Tato množina je rozdělena na množinu X , která obsahuje body s jasovou hodnotou 1 a X^C obsahující body pozadí, tedy s hodnotou 0.

Morfologická transformace [2] je relace mezi množinou X a tzv. strukturním elementem B . Základní a nepoužívanější strukturní elementy jsou zobrazeny na obr. 1.13.

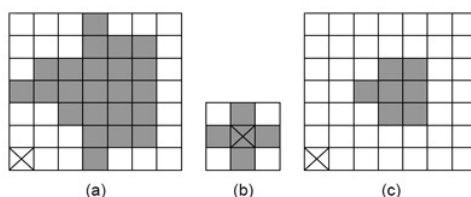


obr. 1.13 – elementy pro binární matematickou morfologii

Binární morfologickou transformaci si lze v postatě představit jako posun elementu B obrazem a pro každý pixel se provede relace mezi X a B . Výsledek je zapsán do výstupního obrazu.

1.6.1 EROZE

Eroze [2] je transformace, při které jsou objekty v obraze zmenšovány. Strukturní element (na obr. 1.14b) se aplikuje na množinu X , tedy v bodech s hodnotou 1. V každém takovém bodě je na obraz přiložen strukturní element. Pokud je element shodný s okolím, je do výstupního obrazu zapsáno 1, jinak 0.

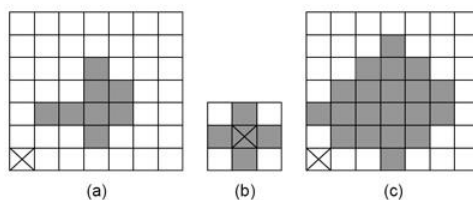


obr. 1.14 – Eroze – (a) vstupní obraz, (b) strukturní element, (c) eroze

1.6.2 DILATACE

Operace dilatace [2] (obr. 1.15) je duální operací k erozi. Objekty jsou touto operací zvětšovány. Tyto operace ale nejsou inverzní, tedy po provedení eroze a následně dilatace nebo opačně výsledný obraz nemusí být shodný s původním.

Strukturní element (na obr. 1.15b) se aplikuje na množinu X , tedy v bodech s hodnotou 1. V každém takovém bodě je na obraz přiložen strukturní element a do výstupního obrazu jsou zapsány hodnoty 1 tam, kde zasahuje.

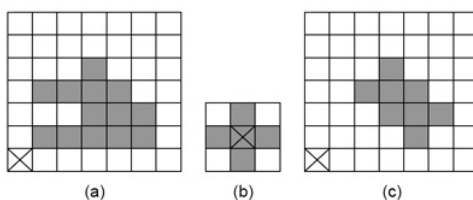


obr. 1.15 – Dilatace – (a) vstupní obraz, (b) strukturní element, (c) dilatace

Operací dilatace jsme schopni odstranit z obrazu díry, které jsou menší než strukturní element.

1.6.3 OTEVŘENÍ

Operace otevření [2] je provedení eroze, následované provedením dilatace. Při provádění obou operací se používá stejný strukturní element. Otevřením dochází k potlačení detailů menších než strukturní element a jsou odděleny objekty spojené přechodem užším než strukturní element.

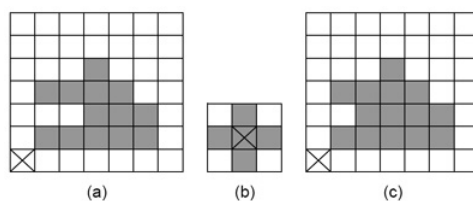


obr. 1.16 – Otevření – (a) vstupní obraz, (b) strukturní element, (c) otevření

1.6.4 UZAVŘENÍ

Tato operace je podobná operaci otevření. V tomto případě však je operace dilatace následována erozí se stejným strukturním elementem.

Uzavřením jsou z obrazu odstraněny detaily menší než strukturní element. Spojí také objekty blízko u sebe a zaplní malé díry v objektech.



obr. 1.17 – Uzavření – (a) vstupní obraz, (b) strukturní element, (c) uzavření

2 KAMEROVÝ SYSTÉM

Očima celého kamerového systému jsou průmyslové kamery, kterými se získává obraz okolního prostředí. Celý systém automatické identifikace obsahuje celkem šest průmyslových kamer, připojených k počítači, který kamery řídí, provádí jejich nastavování a v neposlední řadě vyhodnocuje obraz kamerami pořízený.

Pro zajištění dostatečného osvětlení vagónů v noci jsou použity čtyři výkonné lampy. Každá tato lampa má výkon 400 W a je v nich použita halogenidová výbojka s barevností 80 %. Tyto výbojky jsou použity, protože mají ekonomický provoz a dávají velké množství světla. Dále se vyznačují svou dlouhou životností, nízkým vyzařováním tepla a barevnou kvalitou. Zajišťují osvětlení boční stěny vozu na 30 lx. Takové světlo pro identifikaci postačující, dokonce díky konstantním světelným podmínkám jsou výsledky čtení lepší za tmy než za denního světla.

Z důvodu celoročního venkovního použití je třeba kamery chránit proti chladu, dešti a povětrnostním podmínkám. Jsou použity standardní vyhřívané venkovní kryty kamer VKK 130/230, s napájením 230V, automatickou regulací příkonu dle teploty (až 20W) a stupněm krytí IP 65. Díky tomuto krytu mají kamery optimální provozní podmínky během celého roku a za jakéhokoli počasí.

2.1 PŘIPOJENÍ KAMER

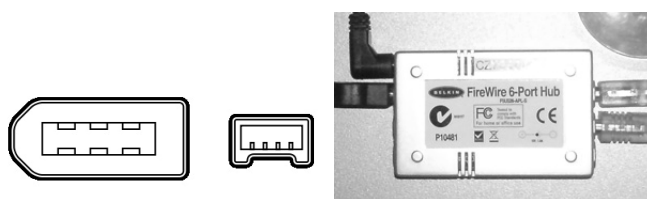
Kamerový systém je centralizovaný pomocí PC. Problém tohoto systému je v tom, že je nutné mít kamery umístěné v určitých vzdálenostech. Z tohoto důvodu je použito různých způsobů připojení kamer. Komunikace s kamerami probíhá na rozhraních *GigE* a *IEEE 1394a*. Oba druhy připojení mají své výhody i nevýhody. Pro připojení digitální průmyslové kamery je pro rozlišení 1024x768, barevnou kameru a 25 fps potřebná rychlost komunikačního rozhraní minimálně 450 Mbit/s. Pro komunikaci s digitální kamerou lze použít více druhů rozhraní. Jsou to například IEEE 1394 (rychlost až 800 Mbit/s), USB 2.0 (pro USB fullrate 480 Mbit/s), GigE (ethernet o rychlosti 1 Gbit/s) nebo Camera Link (rychlost 1,923 Gbit/s). Dále jsou uvedena jen použitá rozhraní.

Firewire a	100 Mbit/s
	200 Mbit/s
	400 Mbit/s
Firewire b	800 Mbit/s
GigE	1 Gbit/s

tab. 2.1 – přehled rychlostí komunikačních rozhraní

Rozhraní IEEE 1394 – Firewire

IEEE 1394 je standardem pro sériovou komunikaci počítače s připojeným periferním zařízením. V současné době je Firewire k dispozici ve dvou verzích. Rozděluje se na Firewire 400 (IEEE 1394a) s maximální rychlostí 400 Mbit/s a Firewire 800 (IEEE 1394b) s maximální rychlostí 800 Mbit/s. Z oblasti připojení průmyslových kamer rozhraní Firewire téměř vytlačilo USB 2.0 s datovou propustností 480Mbit/s. Pro připojení digitálních kamer je ale Firewire praktičtější, protože díky konstrukci svého řadiče (DMA přístup) méně zatěžuje systém. Další výhodou Firewire je možnost napájení kamery přímo z rozhraní, kdy v šestivodičové verzi kabelu je možné dodávat dle normy až 45 W na port při napětí až 30 V.



obr. 2.1 – IEEE 1394 – konektor (6 a 4 pinové provedení), opakovač BELKIN Firewire 6-port HUB

Nevýhodou Firewire je omezení délky kabelu. Tato je omezena z důvodu rušení datového toku okolním prostředím. Norma udává, že maximální délka je limitována kabelem 4,5 m, ale v této aplikaci jsme ověřili možnost použití opakovače signálu, který nám umožnil komunikovat na vzdálenost 14 m (kombinace 10 m a 4,5 m kabelů). Výrobce AVT má ověřeno u svých kamer, že lze použít 10 m kabel, který také výrobce jako nadstandard dodává spolu s kamerou.

Rozhraní GigE

Jedná se o variantu ethernetu s přenosovou rychlostí 1 Gbit/s, která už je plně postačující pro připojení kamer. Původně byl definován pro optická vlákna, ale později byl rozšířen na i na variantu kroucené dvojlinky IEEE 802.3ab, což velice usnadnilo používání tohoto rozhraní.



obr. 2.2 – kabel a koncovka UTP 5e

Při použití kabelu UTP kategorie 5e je definována maximální vzdálenost 100 m. Tato vlastnost umožňuje připojení i vzdálených kamer.

Pro připojení digitálních kamer je možnost rozšíření datového kabelu UTP o napájení. Pokud kamera podporuje napájení přes ethernet, je za připojovacím zařízením (PC, HUB, SWITCH) přidán průchozí zdroj. Tento zdroj je připojen do připojovacího zařízení a samotná kamera je připojena do zdroje. Napájení přes ethernet (Power over Ethernet – PoE) podléhá standardu IEEE 802.3af, napájecí napětí je 48 V s maximálním proudovým odběrem 400 mA. Výkon připojeného zařízení může být v rozsahu 0,44 až 12,95 W. V této aplikaci ale PoE není využito, protože kamery Basler Scout jej nepodporují.

2.2 POUŽITÉ KAMERY

Kamera Guppy F-080C

Kamery Guppy F-080C od firmy Allied Vision Technologies jsou připojené pomocí IEEE 1394a s rychlostí 400 Mbit/s. Kamerový čip CCD od firmy SONY je velikosti 1/3", tj. úhlopříčka čipu je 6 mm. Kamera má maximální rozlišení XGA (1034 x 778), ale je používána v rozlišení 1024 x 768. Rozteč jednotlivých fotoelementů je 4,65 µm x 4,65 µm.



obr. 2.3 – kamera Guppy F-080C

Kamera je barevná s barevnou hloubkou 8 bitů a je schopná snímat až 30 snímků za sekundu. Je napájena přes IEEE 1394a kabel, s napětím 8-30 V a příkon má menší než 2 W. Provozní teplotu má v rozmezí od +5 °C do +50 °C.

Kamera má C-závit vzniklý dodáním 5 mm mezikroužku (přímo od výrobce), při jehož demontáži se jedná o CS-závit s průměrem 25,4 mm. Je proto možnost použít jak C, tak CS objektiv.

Kamera Scout scA1000-30gc

Kamery Scout scA1000 od firmy Basler AG jsou připojeny pomocí rozhraní GigE. Je použit kamerový čip CCD SONY typu 1/3" s maximálním rozlišením XGA (1034 x 778), ale použité rozlišení 1024 x 768. Rozteč jednotlivých fotoelementů je 4,65 μm x 4,65 μm .



obr. 2.4 – kamera Basler Scout scA1000-30gc

Díky připojení přes GigE je možné mít kameru připojenou kabelem až 100 m dlouhým. Kamery Scout ale neumožňují PoE (napájení přes ethernet), proto je nutné vést ke každé kameře paralelně s datovým UTP i napájecí kabel.

Čip CCD

Oba typy použitých kamer používají pro snímání obrazové informace čip typu CCD. Zkratka CCD vznikla z anglického názvu Charge-Coupled Device [12] (česky zařízení s vázanými náboji). CCD čip funguje jako paměťový registr, na který působí dopadající světlo a jeho hodnota je zaznamenána.

Pro snímání barevného obrazu je v kamerách použit jediný CCD čip, využívající principu nanesení barevného filtru na jednotlivé fotocitlivé buňky, které poté zaznamenávají světlo odpovídající jejich filtru. Využívá se bayerovo uspořádání filtru, vycházející z faktu, že je lidské oko nejcitlivější na zelenou barvu. Proto jsou zelené filtry zastoupeny v dvojnásobném množství než červené a modré. Při rozlišení 1024x768 je polovina fotocitlivých buněk se zeleným filtrem, modré a červené jsou zastoupeny po čtvrtině z celkového počtu. Výsledný barevný obraz vznikne kombinací barev aktuálního a okolních buněk.

Zelená	Červená
Modrá	Zelená

tabulka 2.2 – rozložení bayerova barevného filtru

Nevýhodou barevných CCD čipů je jejich menší světelná citlivost.

Řízení kamer

Hlavní rozdíl mezi oběma typy kamer je ve způsobu řízení obrazového zesílení a doby závěrky. Kamery typu Guppy mají funkci automatického řízení doby elektronické závěrky (AES) a nastavení obrazového zesílení (AGC). Funkce jsou vhodné pro jednodušší systémy, kdy má snímaná scéna konstantní nebo jen málo se měnící parametry. V tomto případě nejsou AGC a AEC příliš k užítku, protože regulace probíhá nejprve prodlužováním doby závěrky a poté zvyšováním zesílení. Toto mělo za následek špatné pořizování nočních snímků z důvodu dlouhé doby závěrky, tedy rozmazání snímků díky pohybu objektů v obraze. AGC i AEC jsou tedy nahrazeny programovým řízením s vlastním speciálním algoritmem. Tento algoritmus je stejně nutné použít, protože kamery Scout tyto funkce nemají. U jediné kamery, přesně cam0 (záběr na štěpky), jsou tyto automatické funkce využity.

2.3 OBJEKTIVY

Každá kamera vyžaduje použití speciální optiky, která umožní přesné zaostření světelných paprsků na CCD čip kamery. Tato optika se nazývá objektiv. Objektivy jsou standardizovány a rozděleny podle základních parametrů. Prvním parametrem je způsob připojení ke kameře. Průmyslové kamery mají pro připojení objektivu klasický M závit o určitém poloměru. Hlavní rozdíl je však v tom, jak daleko mají od závitu mají umístěný CCD čip. Rozdělují se podle toho na C a CS kamery. Stejným způsobem jsou děleny i objektivy. Použití stejného typu kamery a objektivu není žádný problém. Ten nastává v případě potřeby kombinovat různé kamery s objektivy. Použití C objektivu na CS kameru je možné s použitím speciálního 5 mm širokého mezikroužku. Použít CS objektiv na C kameru ale nelze. Tyto kombinace jsou uvedeny v tabulce 2.3. Výčet použitých objektivů a kamer v aplikaci je znázorněn v tabulce 2.4, kde jsou uvedeny i informace o pozici kamer.

		Kamera	
		C	CS
Objektiv	C	OK	Redukční kroužek
	CS	X	OK

tab. 2.3 – možné použití objektivů a kamer

Dalším základním parametrem objektivů je vzdálenost ohniska, udávaná v milimetrech. Ohnisková vzdálenost vypovídá o objektivu, jaké přiblížení obrazu provede. Čím větší je ohnisková vzdálenost, tím menší zorný úhel objektiv poskytuje.

účel:	kamera:	objektiv:	ID:
štěpky vykládka	FW – C/CS Guppy F080C (905A)	C – Pentax 12 mm	cam0
Rampa	GigE – C – scA1000 (6B70)	C – Cosmimar 6 mm	cam1
vagony horní	GigE – C – scA1000 (6B71)	C – Kowa 3,5 mm	cam2
vagony dolní	GigE – C – scA1000 (6B72)	C – Cosmimar 6 mm	cam3
nákladňáky - zadek	FW – C/CS Guppy F080C (9058)	CS – Computar 4 mm	cam4
nákladňáky - přede	FW – C/CS Guppy F080C (9059)	CS – Computar 4 mm	cam5

tab. 2.4 – seznam kamer s objektivy a ID

2.4 POPIS ORIENTACE POUŽITÝCH KAMER

Kamera čtení textu z vagónů (cam3):

Pro čtení textu z vagónů byla použita kamera Scout scA1000 s objektivem C Cosmimar 6 mm. Kamera zabírá bok vagónu a obraz z ní je vstupním obrazem pro identifikaci vagónů. Označena jako cam3.

Kamera kontroly nákladu ve vagónech (cam2):

Tato kamera je synchronizovaná s kamerou na čtení vagónů a dokumentuje obsah identifikovaného vozu. Použita kamera Scout scA1000 s objektivem C Kowa 3,5 mm. Značena jako cam2.

Kamera čtení RZ nákladních automobilů (cam1):

Kamera slouží pro identifikaci nákladních automobilů, rozpoznáváním jejich RZ. Použita kamera Scout scA1000 s objektivem C Cosmimar 6 mm. Značena je jako cam1.

Kamera kontroly nákladu aut – přední část (cam5):

Tato kamera je synchronizovaná s kamerou na čtení RZ nákladních automobilů a dokumentuje obsah identifikovaného vozu na korbě. Kamera Guppy F-080C s objektivem CS Computar 4 mm. Značena jako cam5.

Kamera kontroly nákladu aut – zadní část (cam4):

Tato kamera má stejnou funkci jako cam5, jen zabírá příves vozu, který by se nevešel do záběru kamery cam5. Kamera Guppy F-080C s objektivem CS Computar 4 mm. Značena jako cam4.

Kamera štěpky (cam0):

Tato kamera slouží pouze k orientační představě o přítomnosti vagónů se štěpkami. Snímky z této kamery jsou informativní a slouží pouze k archivaci. Kamera Guppy F-080C s objektivem C Pentax 12 mm. Značena jako cam0.



cam3



cam2



cam1



cam5



cam4



cam0

obr. 2.5 – záběry použitých kamer

2.5 SENZORY

Pro rozpoznání jedoucího vlaku jsou použity dvě optické závory, pomocí kterých se určuje přítomnost i směr jedoucího vlaku. Směr lze snadno rozlišit z pořadí signálů z jednotlivých závor.



obr. 2.6 – reflexní závory – instalace

Optosenzor Turck QS30LV – reflexní závora

Jedná se o senzor typu optická závora, s vestavěným vysílačem i přijímačem. Světelný paprsek je namířen na odrazku, kterou je vrácen zpět na přijímač. Objekt je detekován při přerušení světelného paprsku. Výhodou je nutnost instalace pouze jednoho přístroje. Instalace snímače typu vysílač přijímač by byla problematická kvůli kolejím, pod kterými by bylo nutné provést vodiče. S těmito závorami je použita bílá odrazka firmy Turck BRT-84 s průměrem 84 mm.



obr. 2.7 – Optosenzor QS30LV

Podrobnější specifikace reflexní závory je možné nalézt na přiloženém CD.

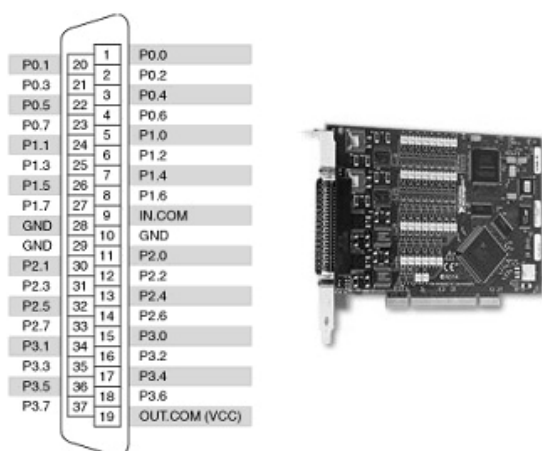
2.6 POUŽITÉ PC

Srdcem celého systému je osobní počítač, ke kterému jsou připojeny všechny kamery, získává informace z čidel a řídí funkci světel. Konfigurace PC (blíže na CD):

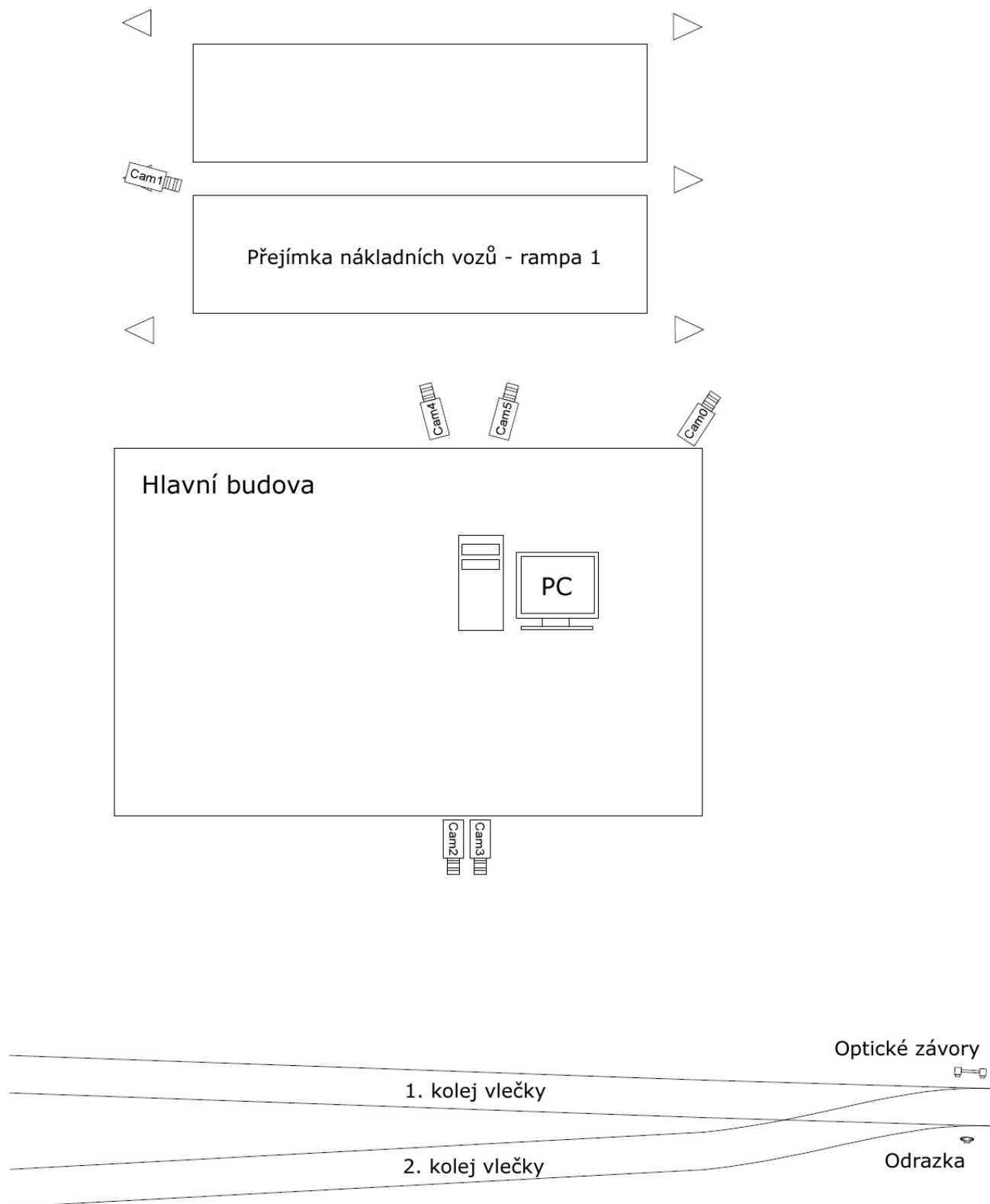
- základní deska – MICROSTAR K9N Platinum
- procesor – AMD Dual-Core Athlon A64 X2 5200+ 64-bit windsdor
- RAM – 2GB ,DDR2 675MHz, PC5400, CL4, CORSAIR
- síťové karty – 2x Intel PRO/1000PT Desk. Adapter Gb Cat-5 cabling, PCIe1x
- grafická karta – ATI (Sapphire) Radeon X300SE, 128 MB DDR
- HDD – SEAGATE Barracuda NL35.2 400GB, SATA II NCQ 8MB cache, 7200ot
- Firewire karty – KOUWELL 582V2 (čip TI) - 3x FireWire, PCI
- DVD – LG GSA-H22L black
- FDD NEC 3.5"/ 1.44MB
- Microsoft Windows XP Professional CZ
- KME MidiTower ATX CX-0762, P4-400W
- PC I/O karta od NI 6519

PC I/O karta od NI 6519

Pro ovládání světel a komunikaci se senzory je použita PC I/O karta od National Instruments NI 6519. Karta má 16 vstupů a 16 výstupů, díky kterým lze komunikovat přímo s průmyslovými senzory a lze jí přímo ovládat akční členy. Napěťové úrovně karty dovolují přímo připojit snímače a relátka s 24 V logikou. Karta je přímo podporována díky ovladačům NI-DAQmx v LabWindows/CVI, takže je pro tuto aplikaci ideální.



obr. 2.8 – PC I/O karta NI 6519



obr. 2.9 – schématický plán kamerového systému

2.7 SOFTWAREOVÁ ČÁST

2.7.1 IDENTIFIKACE KOLEJÍ

Tato část programu se stará o lokalizaci a identifikaci kolejnic. Je důležitá, protože se pomocí ní provádí rozpoznání kolejnic, po kterých se vagón pohybuje. Pokud je na 2. koleji, neprovádí se čtení ID, protože ze strany firmy byl požadavek na identifikaci vagónů přijíždějících na první kolej, kde probíhá vykládka dřeva. Na druhou kolej přijíždějí již vyložené vagóny, případně slouží jako odstavná kolej. Text na snímku z 2. koleje odpovídá pouze 55% textu při snímku z první koleje. Identifikace koleje se provádí proložením cca 30 krátkých jasových profilů *imaqLineProfile()* v místě přibližné polohy kolejnice. V nich je poté nalezena světelná hrana (kolejnice se leskne) a takto vzniklé body se spojí. Tím je získán profil přesně kopírující první kolej. Sledováním změn jasových hodnot při průjezdu vlaku je vyhodnoceno, zda se jedná o vlak na první nebo druhé koleji.

2.7.2 VERIFIKACE ID VAGÓNU

Díky tomu, že formát ID vagónu se řídí podle přesně daných zásad a obsahuje i kontrolní součet, je možné provádět ověření správnosti rozpoznaného textu. Tato knihovna je schopna provést i rekonstrukci poškozeného textu, dokonce může opravit chybně přečtený znak. Příklady opravení textu (symbol ? odpovídá neidentifikovanému znaku):

- přečtený text → verifikovaný text → ID pobytu vozu
- 21545?576730 → 215455576730 → 447808
- 31545??98743 → 315459?98743 → 447708
- ??5?5526994? → **21545526994?** → 447649

Knihovna je svou funkcí obdobou kontroly pravopisu. Vytvoření verifikace však nebylo předmětem mé práce.

Nicméně se na tento způsob nedá vždy 100% spolehnout, protože pokud dojde ke ztrátě znaku (znak není nalezen, dojde k jeho vypuštění z textu), může být tato verifikace i na překážku. Kvůli posunu znaků dochází k zásadní změně v ověřování a tím i k ještě většímu poškození textu. V krajním případě se může stát, že je snímek, přiřazen jinému ID pobytu vozu.

2.7.3 AUTOMATICKÉ PŘIJÍMÁNÍ EMAILŮ

Pro možnosti identifikace přítomných vagónů podle ID pobytu vozu je nutné získávat do systému průběžně nová data s informacemi propojujícími ID pobytu voze s identifikačním číslem na voze. Toto ID pobytu vozu je každému vagónu přiřazeno při průjezdu vstupní kontrolou do areálu papíren. Po rozlišení kvality a druhu obsahu vagónu je dáno místo určení každého vozu. Na místo určení je poté odeslán informační email s daty jednotlivých vozů.

Pro tuto potřebu je vytvořen modul pro automatické přijímání emailu, který po obdržení emailu, provede dešifrování dat a zařadí je databáze příchozích vozů. Každé ID pobytu vozu je platné po dobu 48 hodin, kdy musí odpovídající vůz dorazit.

3 KNIHOVNA NI – VISION

Knihovna NI–Vision je knihovna pro LabWindows, LabView i MS VC++. Obsahuje již hotové základní i složité funkce pro zpracování obrazu, jako barevné operace, kalibraci obrazu, matematickou morfologii a mnoho dalších. V této práci jsou z toho mnoha zmíněny jen některé, které byly použity při tvorbě modulu pro čtení RZ a ID. Pro provoz zkompilovaného programu je nutné použít na cílovém počítači RTE (RunTime Engine) licenci. Celý program vznikl na verzi NI–VISION 8.0, přičemž na použitém PC je RTE ve verzi 8.2.

3.1 DŮLEŽITÉ FUNKCE

Použité důležité funkce (abecední řazení):

Bw_LPF(...)

imaqCopyCalibrationInfo(...)

imaqCorrectCalibratedImage(...)

imaqCreateCharSet(...)

imaqFillHoles(...)

imaqFindPattern(...)

imaqChangeColorSpace2(...)

imaqLearnCalibrationGrid(...)

imaqLearnPattern(...)

imaqLineProfile(...)

imaqLocalThreshold(...)

imaqMatchShape(...)

imaqMorphology(...)

imaqReadOCRFile(...)

imaqReadText2(...)

imaqRejectBorder(...)

imaqResample(...)

imaqSizeFilter(...)

imaqTrainChars(...)

imaqWriteOCRFile(...)

Podrobný popis funkcí se nalézá v příloze č.4

3.2 OCR TRAINING

Jednou ze zásadních funkcí knihovny NI-VISION je funkce OCR. Tato umožňuje číst znaky na přesně definované pozici v obraze. Je samozřejmé, že je potřeba tento text tedy přesně zaměřit a také je nutné, aby funkce pro čtení měla vytvořenou abecedu, ze které při čtení vychází. Pokud tedy čtené znaky v abecedě schází, funkce ho nerozpozná nebo mu přiřadí hodnotu nějakého podobného znaku.

Největším omezením funkce OCR je nemožnost víceřádkového čtení. Proto je nutné této funkci předkládat pouze jednořádkový text. Toto omezení platí například i pro případ, že je nad textem např. nakreslená čára nebo nějaký jiný obrazec. V takové situaci jednotlivé znaky nejsou vůbec zaregistrovány a tudíž nemohou být ani přečteny. Nicméně OCR je v knihovně NI-VISION velice platný a silný nástroj, který nabízí velkou škálu možných nastavení a tím i použití. Blíže o funkci OCR v následující části textu.

3.2.1 ZNAKOVÁ SADA

První věcí, kterou je potřeba udělat při programování funkcí automatického čtení z obrazu, je vytvoření znakové sady. Jedná se tedy o vytvoření seznamu jednotlivých znaků, které chceme, aby automatické čtení dokázalo rozlišit. V této abecedě se také nastavují jednotlivé parametry čtení, jako módy prahování, nastavení rozmezí rozměrů znaků v obou osách i nastavení rozsahu plochy odprahovaného znaku. Dále je zde například možnost nastavení způsobu a přesnosti čtení, nastavení substitučního znaku a úroveň pro přijetí znaku jako identifikovaného. Podrobněji dále v textu.

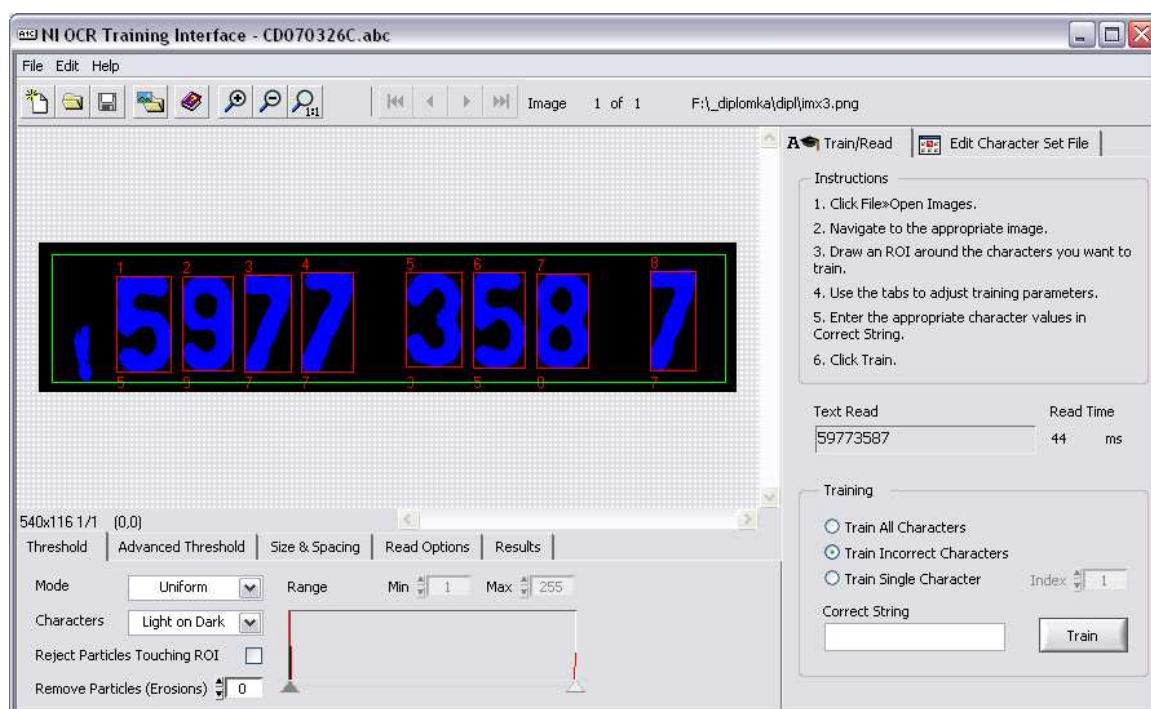
3.2.2 PROGRAMOVÝ ZPŮSOB VYTVÁŘENÍ ZNAKOVÉ SADY

Programově lze soubor s abecedou (struktura *CharSet*) vytvořit pomocí funkce *CharSet* imaqCreateCharSet()*. Funkce vrací pointer na novou (prázdnou) znakovou sadu. Takto vytvořenou sadu je nutné doplnit o jednotlivé grafické předlohy znaků, spolu s čtecími parametry. Toto lze provést pomocí funkce *imaqTrainChars(...)*. Tento způsob je pro jednorázové vytvoření abecedy nevýhodný, ale jeho nesporná výhoda je možnost programového doučování znakové sady. Toto je vhodné například pro programy, kde je vytvořeno komunikační rozhraní pro obsluhu programu a je potřebné doučení např. nového znaku nebo možné doplňování již známých znaků do abecedy. Velmi výhodné bývá naučení jednoho znaku z více různých předloh. Tím se velice zvyšuje pravděpodobnost správného

rozpoznání toho konkrétního znaku. Pro uložení znakové sady se používá funkce *imaqWriteOCRFile(...)*.

3.2.3 VYTVÁŘENÍ ABECEDY POMOCÍ NI OCR TRAINING INTERFACE

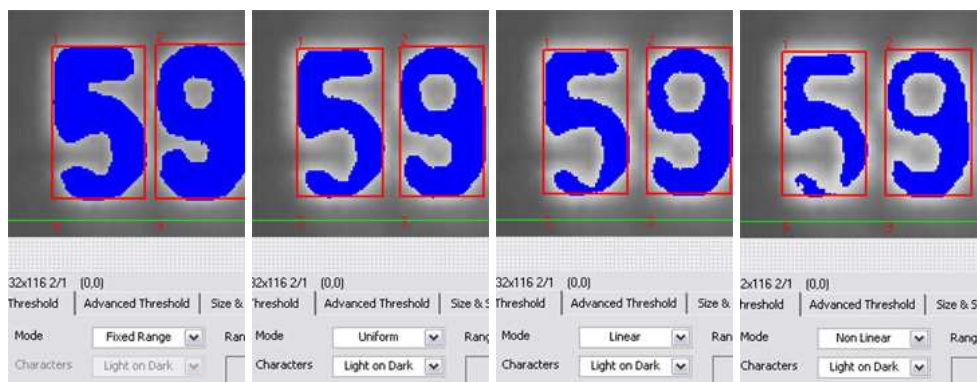
NI OCR Training Interface je prostředí vytvořené firmou National Instruments a automaticky dodávané spolu s knihovnou NI-VISION. Je to programové prostředí usnadňující nastavování parametrů znakové sady a učení jednotlivých znaků. Tyto lze učit buď samostatně nebo jako text celý řádek.



obr. 3.1 – základní okno aplikace NI OCR Training Interface

Na obrázku 3.1 je vidět základní okno aplikace NI OCR Training Interface. Je to klasická aplikace pro Windows, takže ovládání je intuitivní a snadné. Je zde možnost vytvoření nové znakové sady, editace již hotových znakových sad nebo jen odzkoušení, zda je text vůbec čitelný. Pro tuto poslední možnost je možné otevřít víc obrazových předloh najednou a jednoduše mezi jednotlivými obrázky přepínat. První a zásadní možností nastavení znakové sady v záložce Threshold je informace, jestli hledat tmavý text na světlejším pozadí nebo naopak (tedy Characters: Dark on Light nebo Light on Dark). Další možností nastavení vlastnosti čtení je ThresholdMode, neboli mód prahování obrázku. Je to způsob, jakým bude funkce OCR pro vlastní potřebu převádět šedotónový obraz na binární.

Na obrázku obr. 3.1 je text již převeden na binární obraz, proto se jednotlivé módy prahování na tomto obrázku neprojeví. Pro lepší přehlednost možností jednotlivých módů je potřeba použít obrázek šedotónový.



obr. 3.2 – příklady nastavení prahovacího módu (Fixed Range, Uniform, Linear, Non Linear)

Popis jednotlivých prahovacích módů:

Fixed Range – základní mód, kdy se přesně určí rozmezí jasů, které chceme brát v potaz pro prahování pro celý vyhledávací region. Vhodné pro obrazy se stejnými jasovými hodnotami. Výhodou je velká rychlost.

Uniform – funkce, při které OCR samo z regionu určí nejlepší hodnotu prahovací úrovně opět pro celou oblast.

Linear – region pro čtení je rozdělen do několika částí (jejich počet je ručně nastaven). OCR vypočte hodnotu prahovací úrovně pro první a poslední blok. Poté určí pomocí lineární interpolace hodnoty pro bloky mezi nimi.

Non Linear – region pro čtení je rozdělen do několika částí (jejich počet je ručně nastaven). OCR vypočte hodnotu prahovací úrovně pro každý blok zvlášť.

Poslední možností nastavení v záložce Threshold je možnost nebrat v potaz objekty, které neprojdou skrze selekci pomocí opakované eroze obrazu. Počet těchto erozí lze nastavit pod označením Remove Particles.

Další záložkou pro nastavení parametru znakové sady je záložka Advanced Threshold. Tato není aktivní pro nastavení pomocí Fixed Range, ale pro ostatní volby má funkci doplňující nastavení.

obr. 3.3 – záložka Threshold

Pomocí nastavování Lower a Upper Value lze měnit mezní hodnoty, které OCR používá pro výpočet. Hodnota Blocks udává počet bloků pro Linear a Non-Linear metodu výpočtu hodnoty mezního jasu. Zaškrtnutí pole „Optimize for Speed“ znamená zrychlení výpočtů, ovšem za cenu snížení přesnosti. Pokud je zapnuta optimalizace rychlosti, je možné aktivovat i „Bi modal Calculation“. Tato funkce umožňuje rychleji počítat jak horní, tak dolní mez jasu u obrazu, které jsou reprezentovány dvěma většinovými hodnotami jasu. Jako mezní hodnota pro počet pixelů určitého jasu je brána minimální velikost znaku, nastavená v záložce Size&Spacing.

Záložkou pro nastavování velikosti znaků je záložka Size&Spacing. Zde lze nastavit maximální a minimální rozměry jednotlivých znaků. Je možné nastavit minimální mezeru mezi jednotlivými znaky a minimální a maximální plochu, kterou znak zabírá.

obr. 3.4 – záložka Size & Spacing

Pomocí funkce AutoSplit lze od sebe oddělit i znaky, které jsou nakloněné a vzájemně si zasahují do průmětů. Hodnota Max. Element Spacing udává, jaká musí být mezera mezi jednotlivými sousedními elementy, v horizontálním(x) a vertikálním(y) směru.

Pro nastavení metod čtení slouží záložka Read Options. V této záložce se nastavuje „Acceptance Level“ určující, zda ohodnocení pravděpodobnosti nalezeného znaku je pro čtení přijatelné, tedy zda je znak rozpoznán nebo ne. Pokud znak byl zjištěn, ale nebyl rozpoznán, je v přečteném textu nahrazen symbolem, který se zadává pod označením „Substitution Character“. Také se zde volí mezi dvěma základními volbami strategií čtení – Conservative a Aggressive.

obr. 3.5 – záložka Read Options

Conservative (konzervativní) – používá širokou škálu porovnávacích kritérií pro určení shody testovaného znaku a jednou z předloh ze znakové sady.

Aggressive (agresivní) – tento způsob vyhodnocení shody porovnávaného znaku s předlohou využívá pouze několik málo porovnávacích kritérií. Má však jednu zásadní přednost. Je mnohem rychlejší než konzervativní kritérium.

Kromě těchto porovnávacích kritérií lze rychlost, ale také kvalitu čtení ovlivnit pomocí nastavení rozlišovací schopnosti kritéria. Je možné nastavit tři úrovně (High, Medium, Low), které udávají s jak velkými detaily bude vyhodnocovací kritérium pracovat. Poslední možností zde nastavitelnou je „Aspect Ratio Dependent“ aktivující políčko „Aspekt Ratio“, které udává maximální možnou změnu rozpoznávaného znaku a naučené předlohy. Toto rozmezí lze nastavit od 100 do 400%.

V obou případech použití OCR je využíván konzervativní mód s vysokou (High) úrovní rozlišování, protože je třeba přesnějšího čtení.

Poslední záložkou je záložka „Results“, tj. výsledky čtení. Zde se udává pořadí identifikovaného znaku, jemu přiřazená hodnota (přečtený znak). Dále jeho pozice, rozměry a velikost. Poslední a nejdůležitější informací je klasifikační skóre, které může nabývat hodnot v rozmezí 0..1000 a udává úroveň shody mezi přečteným znakem a originálem.

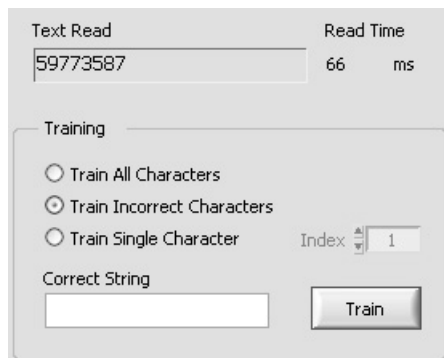
Threshold	Advanced Threshold	Size & Spacing		Read Options		Results		
Character	Class	Left	Top	Width	Height	Size	Class Score	Ident. Score
1	5	58	27	42	71	1798	753	
2	9	106	27	42	71	2051	839	
3	7	156	27	35	72	1229	988	
4	7	202	25	37	72	1251	986	

obr. 3.6 – tabulka výsledků čtení Results

Učení znaků

Přidávání znaků do abecedy lze provádět jednotlivě, tj. po jednom znaku, a to právě označenému znaku se přiřadí hodnota, podle které je zařazen do seznamu naučených znaků.

Další možností je naučit celé slovo. Toto pak automaticky přiřadí hodnoty nalezeným předlohám. Počet učených znaků musí odpovídat počtu nalezených předloh.



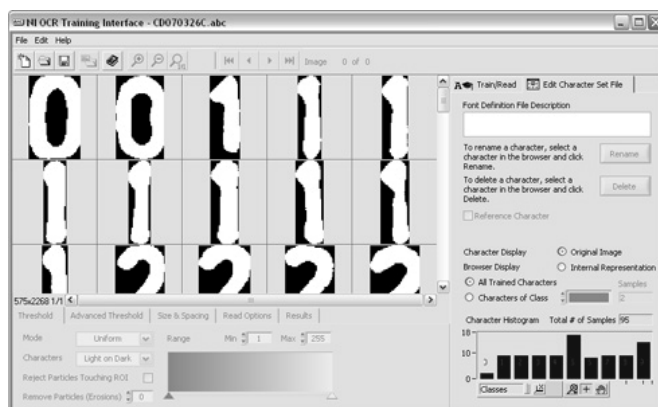
obr. 3.7 – způsob učení znaků

Poslední možností je „Train Incorrect Characters“, tedy učení pouze nerozpoznaných znaků. Pokud požadujeme, aby abeceda obsahovala od každého jen jeden znak, použijeme tuto možnost. Z celého textu použitého pro učení, jsou do abecedy přidány jen znaky, které zatím nebyly naučeny.

Pod položkou „Text Read“ je zapsán rozpoznaný řetězec ve znění, jak ho ORC identifikovalo. Za ním je dále uveden čas, jaký celé čtení trvá.

Editace abecedy

Důležitou věcí ve správě znakové sady je editace abecedy („Edit Character Set File“). V této části NI OCR Training Interface je možné provádět správu již naučených znaků.



obr. 3.8 – záložka editace abecedy

Je zde možnost překlasifikovat špatně naučené znaky, je možné znakovou abecedu zbavit nevhodných obrazových předloh a také je možné nastavit vybraný znak jako referenční. V tomto případě je potřebné nastavit referenční znak v každé rozpoznávané třídě. Při čtení textu je poté text porovnáván pouze s těmito referenčními znaky. Pokud je referenční

znak přejmenován, tj. přeřazen do jiné třídy, je automaticky zbaven reference. Tuto je nutné znovu přidělit.

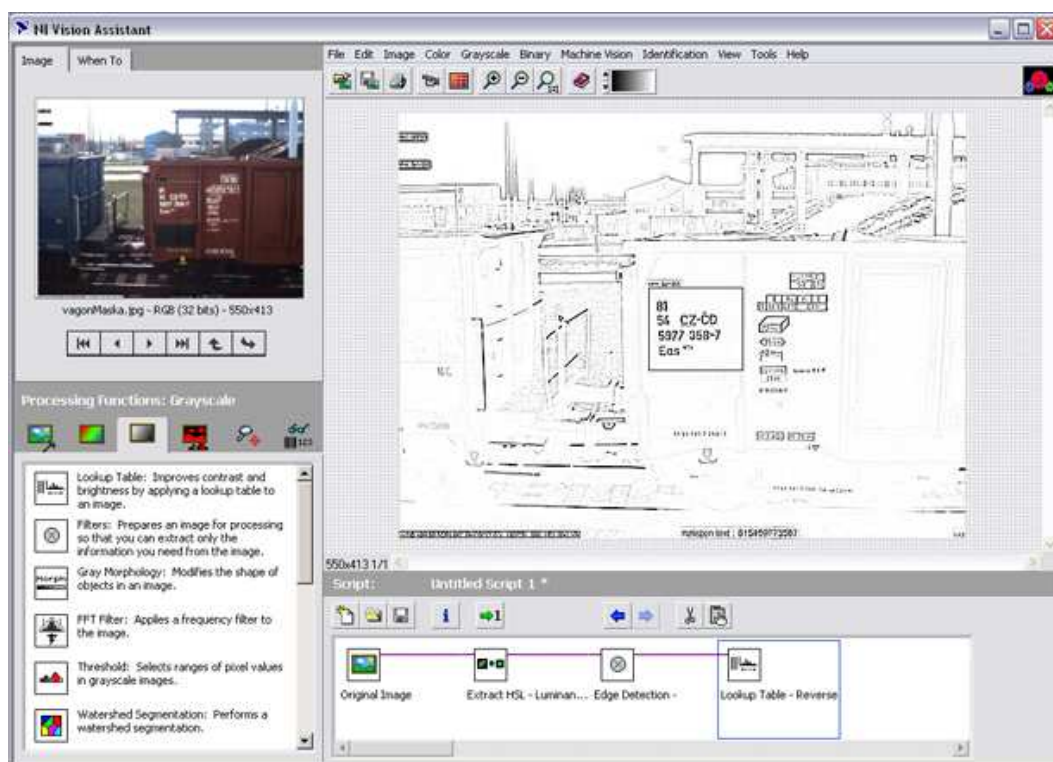
Je také možné zobrazit originály znaků, tj. obrázky, ze kterých byl znak učen, a také reprezentaci jednotlivých znaků pro potřeby výpočtu OCR. Pro zobrazení nikoli všech naučených znaků, ale pouze znaků naučených pro jednotlivé třídy, se použije pole „Characters of Class“. Je to vhodné pro rozsáhlé znakové sady pro větší přehlednost při opravách. Poslední důležitou informací je celkový počet naučených znaků a jejich počet pro jednotlivé třídy.

3.2.4 Použití OCR

Pro načtení již hotové znakové sady z konkrétního umístění slouží funkce *imaqReadOCRFile(...)*. Po přesném vymezení oblasti s textem se provede čtení pomocí funkce *imaqReadText2(...)*. Návratovou hodnotou této funkce je datová struktura *ReadTextReport2*. Výsledek je předán jako pole. Struktura obsahuje informace o počtu přečtených znaků, aktuální přečtený znak a jeho vlastnosti. Mezi nejdůležitější vlastnosti patří skóre shody a přesné souřadnice znaku.

3.3 VISION ASSISTANT 8.0

Součástí programovacího prostředí s knihovnou NI-VISION je aplikace Vision Assistant 8.0, která usnadňuje návrh jakékoli aplikace pracující s obrazem. Je zde možnost vytvoření posloupnosti obrazových operací a tu pak postupně používat na sadu snímků. Tyto snímky lze do sady přidat postupně nebo je možné přidat celý adresář.



obr. 3.9 – okno aplikace Vision Assistant 8.0

V levém horním rohu Vision Assistantu (obr. 3.9) je ovládání nahrané obrazové sady. Je možné přejít po jednom snímku vpřed a vzad, skok na konec sady nebo spustit postupné zpracovávání jednotlivých snímků.

V levém dolním rohu jsou jednotlivé záložky (obr. 3.10) pro postupné zpracovávání obrazu. Zde jsou jednotlivé funkce řazeny podle postupného zpracovávání obrazu pro lepší přehlednost.



obr. 3.10 – záložky pro přehlednost funkcí

Tyto záložky jsou obrazové, pro barevné zpracování, šedotónové zpracování, binární operace, strojové zpracování obrazu a poslední záložkou je identifikace (OCR, OCV). K jednotlivým funkcím lze přistupovat i přes nabídkové menu.

Vytvořený script (posloupnost použitých funkcí) je možné uložit a poté znovu načíst. Je zde možné měnit pořadí použitých funkcí a různě měnit jejich nastavení i poté, co jsou již zařazeny ve scriptu. Celý script je možné nechat provést automaticky. Pouhým označením jedné funkce v pořadí scriptu se zajistí provedení pouze k této funkci. Je tím možné snadno odladit nastavení jednotlivých funkcí v již hotovém scriptu. Každému scriptu je také možné napsat popis, pro lepší pozdější orientaci.

V levém horním rohu je záložka „When to“, která zpřístupňuje základní nápovědu k jednotlivým funkcím v použité paletě. Je zde jejich popis a doporučené použití. Také zde lze nalézt podmínky použití jednotlivých funkcí.

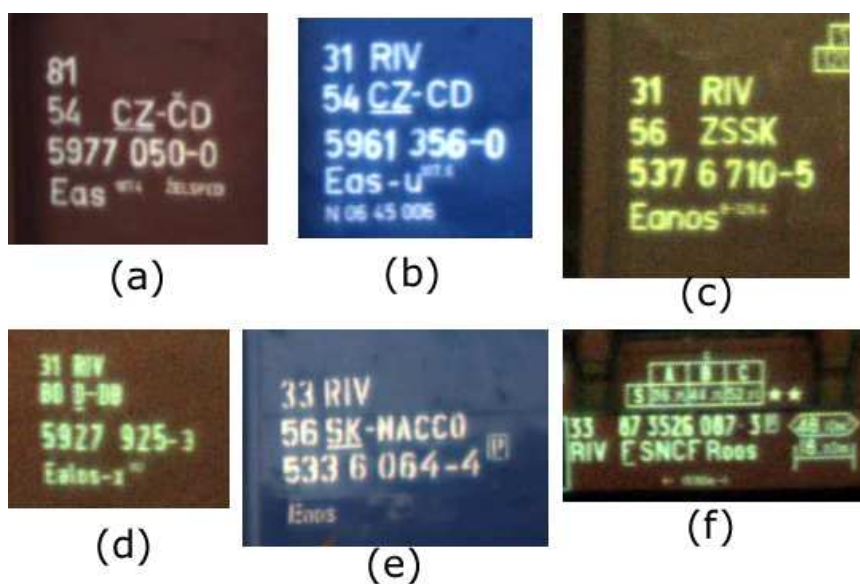
Algoritmy Vision Assistant 8.0 jsou založeny na funkcích z knihovny NI-VISION v LabWindows/CVI. Je proto možné snadno naprogramovat vytvořený a odzkoušený script. Kromě této možnosti je od verze Vision Assistantu 8.5 podporována možnost generování C kódu i kódu pro LabView.

4 MODUL ČTENÍ DRÁŽNÍCH VAGÓNŮ

Modul čtení drážních vagónů je jednou z hlavních částí celého systému pro automatickou identifikaci drážních vagónů. Vstupními daty pro modul je obraz pořízený nadřazeným systémem. Po zpracování tohoto obrazu je nadřazenému systému vrácen řetězec nalezených znaků, jejich pravděpodobnost a pozice textu v obraze.

4.1 OMEZENÍ IDENTIFIKACE

Z důvodu velkého množství různých druhů nákladních vagónů bylo přikročeno k dohodě se zadavatelskou firmou na omezení druhů vozů, které bude systém schopný automaticky identifikovat. Toto omezení se dotýká vozů, které mají nejmenší vytíženost a nebo jejich parametry neumožňují spolehlivou identifikaci. Největším problémem je použití různých fontů (druhů písma) na vozech z různých států. Dalším zásadním omezením identifikace je umístění ID na klaničácích, které je umístěno v místech, kam není kamera namířena. Pro identifikaci tohoto druhu vagónů by bylo zapotřebí další kamery, což je vzhledem k nízké frekvenci průjezdů neekonomické.



obr. 4.1 – různé fonty vagónů (a – český uhlák, b – český modrý uhlák, c – slovenský uhlák, d – německý uhlák, e – slovenský modrý uhlák, f – německý klaničák)

Pro velkou odlišnost, malé rozměry textu a nízký počet průjezdů takto značených vagónů, bylo rozhodnuto německé vozy neidentifikovat (obr. 4.1 d, f). Celkový průjezd těchto vozů za duben 2008 byl 33 vozů z 1375 všech variant (tedy 2,4 %).

Pro řešení dalších odlišných vozů (obr. 4.1e), modrých slovenských uhláků, je vytvořena další větev modulu. Toto bude popsáno dále v textu.

4.2 POPIS MODULU

Modul čtení drážních vagónů je do celkové aplikace přidán pomocí dvou hlavičkových souborů a dvou zdrojových souborů.

- vagonyOCR.h
- nalezeni_vagonu.h
- vagonyOCR.c
- nalezeni_vagonu.c

Kromě hlaviček funkcí jsou v hlavičkových souborech definovány konstanty pro pozdější modifikace modulu a definici konstant pro řízení překladače. Celý modul se volá jedinou funkcí *cteni(...)*, která vrací celočíselnou hodnotu. Tato hodnota vrací informaci o průběhu funkce. Blíže je funkce *cteni(...)* popsána dále. Od nadřazeného systému je vyžadován vstupní obraz pořízený kamerou cam3, a pole vyhledávacích masek. Kromě již zmíněné návratové hodnoty je výstupem přečtený text, jeho identifikační skóre a hodnota udávající polohu nalezeného textu v x-ové ose.

Postup zpracování obrazu je dále popsán v této kapitole.

4.2.1 ROZLIŠENÍ MODRÝCH VAGÓNŮ

Po rozšíření modulu čtení o slovenské modré vagóny bylo za potřeby tyto vozy odlišit. Tato separace probíhá jednoduchým způsobem. Na začátku celého zpracování je provedeno převedení barevného prostoru malého výřezu v pravé části obrazu z RGB na HSL.

Tento převod se provádí postupným převedením jednotlivých pixelů mezi barevnými prostory pomocí funkce *imaqChangeColorSpace2(...)*. Při převádění výřezu se zároveň získává histogram složky *Hue*. Z tohoto histogramu se poté získá ve výřezu převládající hodnota barevnosti. Podle této hodnoty se poté vyhodnotí, jestli je barva obrazu v intervalu definovaném jako modrá. V hlavičkovém souboru *vagonyOCR.h* jsou meze definovány jako interval $\langle \text{modra_dolni}, \text{modra_horni} \rangle$ kde hodnota 255 odpovídá 360°.

Pokud je přítomný vagón vyhodnocen jako modrý, je spuštěna zvláštní větev identifikace, identifikace slovenských modrých. V tomto případě se postupuje obdobně jako

při čtení ostatních vozů (viz dále), jen s tím rozdílem, že pro lokalizace textu probíhá rovnou druhým způsobem (kapitola 4.2.2.2) a jsou při ní použity jiné rozměry hledaného prostoru textu. Po lokalizování ID vozu je provedeno přečtení textu. Je samozřejmě použita speciální abeceda.

Protože může nastat situace, že je vůz modrý, ale není slovenský (ČD používají modrou variantu velkého uhláku), provádí se dodatečně ještě identifikace jako u normálních typů. K této variantě dochází, pokud nebyl modrý vagón identifikován nebo pokud průměrná hodnota identifikačního skóre textu nepřesáhne 350, tedy 35 %.

Pokud nebyl vagón identifikován jako modrý, probíhá klasický způsob identifikace. Prvním problémem je lokalizace textu v obraze. Tomuto se věnuje další podkapitola.

4.2.2 PROBLÉM LOKALIZACE TEXTU

Jedním z hlavních problémů celé práce byla lokalizace identifikačních znaků vagónu v obraze. K nalezení textu na bočnici vagónu jsou používány dva různé přístupy, které jsou v následujícím textu popsány. Dva různé způsoby vyhledávání jsou v programu začleněny z důvodu rozšíření požadavků. Nejprve bylo požadováno hledání ID pouze na první bočnici vozu s omezením pouze na malý a velký uhlák. Po rozšíření požadavků na hledání i na druhé bočnici, byl vyhledávací algoritmus rozšířen. Lokalizace tedy probíhá ve dvou fázích. Nejprve probíhá univerzální lokalizace a pokud se nepodaří nalézt text, provede se hledání pouze na první bočnici.

4.2.2.1 1. ZPŮSOB LOKALIZACE – UNIVERZÁLNĚJŠÍ ZPŮSOB

Tento způsob vychází z hledání textu pomocí předloh, které jsou nalezeny funkcí *imaqFindPattern(...)*. Tímto způsobem se provede vyhledání pěti různých masek a jako referenční bod se bere nejpravděpodobnější shoda, vyjádřená pomocí score v rozmezí 0 .. 1000, kde 1000 odpovídá shodě 100%. Tyto masky jsou pro urychlení celého výpočtu předávány hlavní funkci *cteni(...)* jako pole pointerů na jednotlivé masky. Jejich počet je možné dynamicky měnit pro potřeby budoucích úprav programu. Toto předání se provádí z jednoduchého důvodu. Funkce *imaqFindPattern(...)* využívá pro vyhledání předlohu,

na které již bylo provedeno parametrizování obrazu pro vyhledávání pomocí funkce *imaqLearnPattern* (...). Vyhledávání se samozřejmě provádí jen ve vymezené části obrazu. Je to kvůli úspoře času, tj. prohledávání menšího prostoru, ale hlavně se hledá jen tam, kde se text může nalézat.

Zde jsou získány parametry pro hledání jak v posunu, tak v rotaci. Tato parametrizace je však výpočetně značně náročná, a proto se neprovádí pro každý obraz vstupující do funkce *cteni*(...), ale pouze jednou, při startu celého programu. Dochází tím k významné úspoře strojového času.



obr. 4.2 – Masky pro vyhledání textu

Návratovou hodnotou fce *imaqFindPattern*(...) je struktura *PatternMatch* obsahující informace o středu nalezené shody, rotaci, měřítku, skóre a informace o rozích výběru. Podle získaných parametrů se vyhodnotí, která maska odpovídá nejlepšímu skóre. Parametry nejlepší masky jsou poté brány jako reference pro následné čtení textu.

Vyhledávání probíhá v šedotónovém obrázku, protože zde vyvstává problém s binarizací celého obrazu v důsledku různých hodnot jasu a kontrastu v rámci různé denní doby.



obr. 4.3 – nalezený text a označený výběr pro čtení

Jednou z návratových hodnot funkce *ctení(...)* je i informace o masce, která byla pro lokalizaci použita. Je to z důvodu možné obměny masek po vysledování např. nejméně používané masky.

4.2.2.2 2. ZPŮSOB LOKALIZACE

Tento způsob byl původním, tj. vyhovující počátečním podmínkám a omezením projektu. Nejprve se provedla lokalizace prvního čela vagónu, na kterém se pokračovalo hledáním textu. Po zavedení lokalizace pomocí hledání masek tento způsob neztratil svůj smysl. V případě, že selže první způsob, provede se hledání ještě tímto způsobem. Tento případ může nastat i v případě, že je text přítomen, pouze nebylo dosaženo požadované skóre ani jednou z masek. Tato hranice je nastavena v hlavičce *vagonyOCR.h* jako *potrebne_score* a je přednastavena na 60% (tedy hodnota 599).

První operace s obrazem je převedení snímku na šedotónový. Převodní hodnoty lze nastavit v hlavičkovém souboru *nalezeni_vagonu.h*. Jako první informace se ze snímku získává údaj o tom, zda je snímek pořízen za denního světla nebo za umělého osvětlení. Toto se zjistí pomocí fce *imgLineProfile(...)*, ze které testujeme střední hodnotu jasu v prvním

řádku snímku. Pokud je pod rozhodovací úrovní, je snímek považován za pořízený ve tmě. Na tomto záleží v další části, tj. hledání rozměrů vagónu.

Hledání kraje vagónu

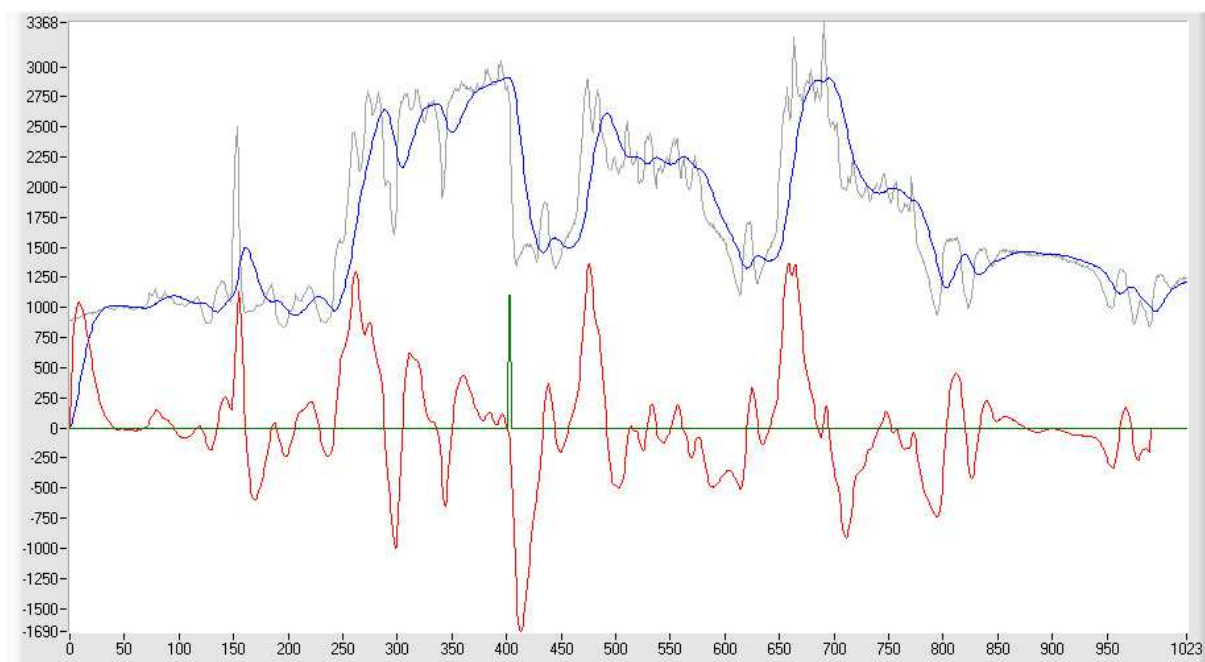
První informace o poloze vagónu, která se zjišťuje je levá hrana, neboli čelo. Toto hledání probíhá tak, že se z obrazu získají hodnoty pomocí fce *imgLineProfile(...)*. Tyto profily se prokládají horizontálně obrazem a jsou sledovány jasové hodnoty pro každý bod profilu. Těchto profilů se obrazem proloží 22 (viz obr. 4.4) a je určována střední hodnota každého bodu pomocí aritmetického průměru. Získaná jasová funkce se poté filtruje pomocí Butterworthova dolnoproustního digitálního filtru, pro odstranění šumu profilu.



obr. 4.4 – Označení pozic jednotlivých jasových profilů a nalezená hrana

Z odfiltrovaného profilu jasového průběhu je dále zjištěna jeho derivace. Díky výraznému jasovému skoku mezi mezerou mezi vagóny a krajem vozu, je pro určení této pozice možné hledat nejvýraznější hodnotu derivace. Přitom je brána jako nejvýraznější kladná pro hledání ve tmě (přechod tma-světlo), protože umělé osvětlení dostatečně nasvítí bok vagónu oproti tomu, že pozadí je neosvětlené, tedy tmavé. Pro obrázky pořízené za denního světla je pro určení pozice prvního čela vozu brána nejvýraznější záporná derivace.

Protože pozadí je dostatečně osvětlené, kdežto bok vagónu je tmavší (přechod světlo-tma). Tato situace je znázorněna na obr. 4.5.



obr. 4.5 – hledání hrany vagónu

šedá – hodnoty jasové funkce získané z profilů (*imaqLineProfile*)
modrá – odfiltrovaný jasový profil
červená – derivace jasového profilu (desetinásobné zvětšení)
zelená – označení nalezené hrany

Největším nedostatkem tohoto způsobu hledání jsou okamžiky mezních světelných podmínek, jak např. východ a západ Slunce nebo zataženo před bouřkou. Také dochází k chybám, pokud jsou na vagónu bílé jistící popruhy nebo vagón poškozený (např. pokreslený grafity). V těchto případech může nastat chybné nalezení hrany. Tyto případy jsou ale ošetřeny 1. způsobem lokalizace, kdy tyto světelné a obrazové podmínky nehrají tak zásadní roli. Může také nastat případ, kdy se v obraze mezera mezi vozy nevyskytne. V tomto případě je hledání vyhodnoceno jako absence mezery a další pokračování v hledání a čtení textu pozbývá smyslu. Pak je návratovou hodnotou funkce *cteni(...)* oznámen tento stav nadřazenému programu.

Hledání výšky vagónu

Pokud se podařilo nalézt levý okraj vagónu, pokračuje lokalizace textu tím, že se zjišťuje výška vozu. Prvním možným způsobem určení této výšky je hledání horní hrany vozu

v místě známého pozadí, jako například v místě stojícího bílého sloupu v pravé části záběru (viz. obr. 4.6). Tato metoda se po odzkoušení v praxi ukázala jako nespolehlivá, protože se často stává, že vagóny jsou naložené vysoko přes horní okraj vozu (tzv. nástavba). Proto jsem změnil určování výšky podle boční hrany, kde nemůže dojít k výskytu nějaké překážky v hledání. Navíc je zde jasová stálost, tudíž je možné hledání nastavit jemněji, tedy i přesněji. Pokud se hrana hledala u sloupu, mohlo dojít k situaci, že se v jasovém profilu objevil text. To vedlo ke vzniku fiktivních hran a čtení pak probíhalo ve špatném prostoru.

Samotné hledání probíhá takto: Nejprve se získají data jasových funkcí pomocí funkce *imaqLineProfile()*. Délku profilu lze definovat v hlavičce *nalezeni_vagonu.h* jako *SirkaLP*. Z této šířky se vychází při určování začátků jasových profilů. Tyto profily jsou situovány v ose x o korekci 20 pixelů napravo od nalezené hrany. V případě, že je hrana nalezená více vpravo, než 600 pixelů, provádí se ještě přepočít této korekce podle funkce $k(x) = 0,77 \cdot (k - 600) - 0,023 \cdot (k - 600)^2 + 5$.

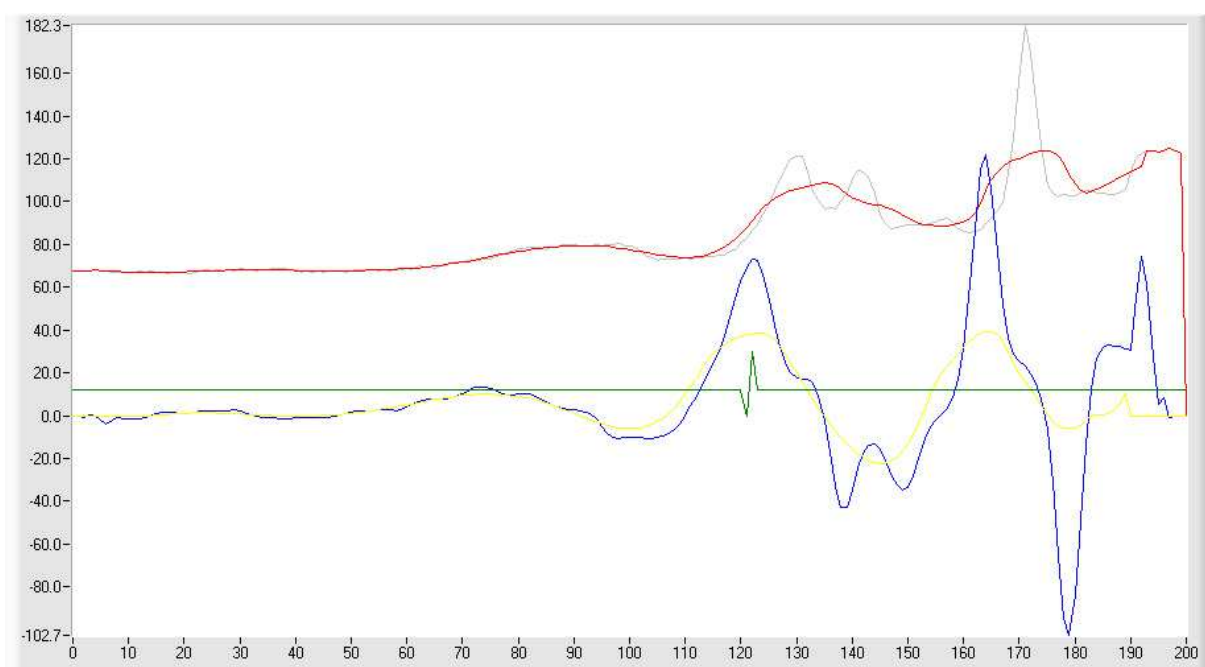


obr. 4.6 – nalezení výšky vagónu

inversní oblast – označení místa, odkud jsou
získávány hodnoty jasové funkce
červená přímka – nalezená horní hrana na voze
označení pozice bílého sloupu

Počáteční souřadnice v ose y obrazu se získá jako daná hodnota maximální výšky horního lemu a k této je přičtena definovaná délka jasového profilu. Celkem se získají údaje

ze dvaceti jasových profilů, kdy rozdíl aktuálního proti předchozímu je posun o jeden pixel vpravo. Z těchto se opět udělá aritmetický průměr pro jednotlivé jasové body, se kterým se poté dále pracuje. Nejprve se provede jednoduchá filtrace průměrováním okolních hodnot, abychom získali hladší funkční průběh. Z tohoto průběhu zjistíme jeho derivaci, na které také použijeme prosté průměrování. Hranici lemu vagónu opět poznáme podle maxima derivace. Jen se v tomto případě nesleduje globální maximum, ale první výrazné maximum derivace, při kterém je významně překročena průměrná hodnota jasové funkce. Pokud je taková souřadnice nalezena, další prohledávání končí, protože v prázdném pozadí mohou být i výraznější jasové změny.



obr. 4.7 – hledání horního lemu vagónu

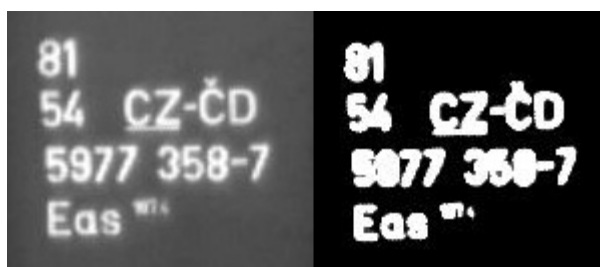
šedá – hodnoty jasové funkce získané z profilů (*imaqLineProfile*)
červená – odfiltrovaný průběh jasové funkce
modrá – derivace jasového profilu
žlutá – odfiltrovaný průběh derivace
zelená – označení nalezené hrany

Určení pozice textu

Nalezené hodnoty rozměrů vagónu jsou poté předány dále programem části pro čtení textu. Tato varianta hledání textu je naprogramovaná samostatně v souboru `nalezeni_vagonu.c`. Je to pro zpřehlednění a možnou implementaci pouze této funkce samostatně, např. pro ověřování přítomnosti vagónu a jeho správné pozice pro čtení identifikačních znaků.

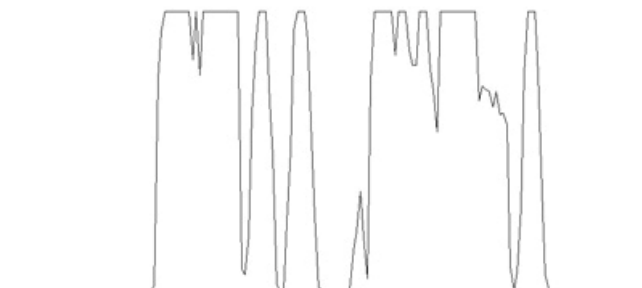
4.2.3 SEGMENTACE TEXTU

Po získání informací o umístění textu, nastává fáze čtení. Prvním, co se provádí je zjištění přesných rozměrů textu. První operací pro určení těchto informací je provedení procentuální prahování obrazu podle histogramu. Tato metoda je podrobněji popsána v kapitole 1.4.3. Hodnota procentuálního prahu je brána podle velikosti výběru s textem. Pokud byl text lokalizován prvním způsobem lokalizace (kapitola 4.2.2.1) je tato hodnota předdefinována na 10 %, pokud byl použit druhý způsob (kapitola 4.2.2.2) je tato prahová hodnota předdefinována na 7,5 %.



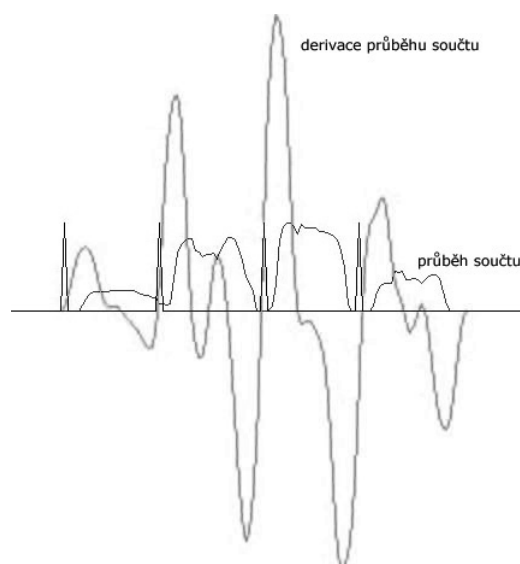
obr. 4.8 – prahovaný obraz (10% nejjasnějších bodů)

Po získání binárního obrazu je provedeno sečtení všech hodnot pixelů podle osy x i podle osy y . Z průběhu součtu podle osy x (obr. 4.9) je určena šířka nejdelšího řádku. Tato šířka je po získání souřadnic řádků v ose y použita pro určení výběru pro třetí řádek textu.



obr. 4.9 – průběh součtu podle osy x (šířka textu)

Z průběhu hodnot součtu v ose y (obr. 4.10) je určována šířka a pozice jednotlivých řádků textu. Nejprve je zjištěna maximální hodnota tohoto průběhu, která odpovídá nejdelšímu, tedy třetímu řádku textu. Poté je sledován průběh derivace z odfiltrovaného průběhu součtu (filtrace průměrováním okolních hodnot). Když tato derivace prochází nulou a zároveň je tento bod minimem v průběhu součtu, je tento bod vyhodnocen jako mezera meziřádky.



obr. 4.10 – průběh součtu podle osy y

(zobrazena i derivace tohoto průběhu a naznačení nalezených řádků)

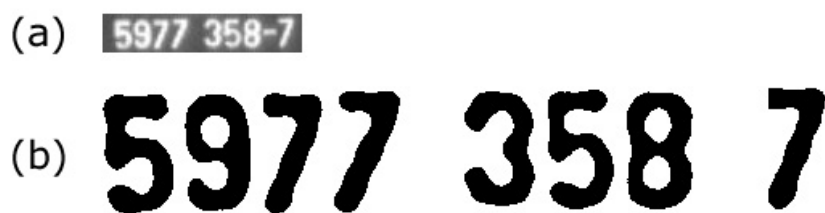
Po určení pozice a rozměrů řádků se přechází do fáze čtení.

4.2.4 ČTENÍ TEXTU

Funkce OCR je v knihovně NI-VISION realizována pomocí funkce *imaqReadText2(...)*, která byla zmíněna již v kapitole 3.2.3. Pro úspěšné přečtení textu je ale nezbytné mít načtenou správnou znakovou sadu, tedy takovou, která byla vytvořena pro čtený font. Pokud se použije jiná abeceda, může se stát, že znaky nebudou vůbec identifikovány, pokud budou, jejich identifikační skóre bude malé. Proto je potřebné rozlišovat formát čteného textu a pro různé varianty písem, vytvářet vždy unikátní znakové sady.

Jsme ve fázi, kdy máme přesné informace o poloze a velikosti jednotlivých řádků s textem. Samotné čtení je realizováno postupně od třetího řádku. Před voláním funkce *imaqReadText2(...)* je nejdříve provedeno čtyřnásobné zvětšení řádku. Díky tomuto zvětšení je získána několikanásobně větší pravděpodobnost správného rozpoznání znaku díky zvětšení jeho celkové plochy, ale také zvýraznění detailů. Funkce pro změnu rozměrů obrazu je *imaqResample(...)*. Při tomto zvětšení je použita aproximace jasových hodnot Bikubická interpolace (viz kapitola 1.5.7). Je to metoda nejpomalejší, ale zato má nejlepší výsledky. Dokud toto zvětšování nebylo realizováno, byla úspěšnost čtení textu výrazně horší.

Takto zvětšený obraz řádku je poté odprahován pomocí *imaqLocalThreshold(...)* s použitím inteligentního lokálního prahování. Po načtení znakové sady je možné provést čtení takto připraveného obrazu (viz obr. 4.11).



obr. 4.11 – třetí řádek textu (a – před zvětšením a prahování, b – upravený řádek v negativu)

Po přečtení tohoto obrazu se postupuje obdobným způsobem pro druhý a následně i pro první řádek. Přečtené hodnoty i s identifikačním skóre pro každý znak jsou předány formou textového řetězce a pole celočíselných hodnot nadřazenému systému.

Po úspěšné identifikaci vagónu je funkce *cteni(...)* ukončena s návratovou hodnotou 0.

4.3 STRUKTURA MODULU

Modul pro automatickou identifikaci drážních vagónů je obsluhován pomocí jediné funkce. Touto funkcí je funkce *cteni(...)*, jejíž přesná hlavička bude v této kapitole popsána. Čtenář zde nalezne i popis možných návratových hodnot této funkce. Kromě funkce *cteni(...)* je pro vnější volání určena ještě funkce *hledej_vagon(...)*, jejíž popis je zde také. Tato funkce je určena pro nalezení mezery mezi vagóny.

Funkce *cteni(...)*

Funkce *cteni(...)* je volána nadřazeným systémem. Je to hlavní funkce modulu čtení drážních vagónů.

Hlavička funkce:

*int cteni (Image *im_in, char *out_text, int *prvdp, int *hrana, Image* mask[], int maskNR);*

im_in
 – vstupní obrázek
 – pro množství informací zaznamenaných do obrazu slouží příkazy pro překladač
out_text – textový výstup (délka řetězce – min 13) obsahující ID čteného vagonu
prvdp – pole hodnot pravděpodobnosti jednotlivých znaků v ID
hrana – x-ová souřadnice nalezené hrany vagonu
mask[] – pole masek pro lokalizaci textu
maskNR – počet masek pro lokalizaci textu

– návratové hodnoty
 0: vše proběhlo v pořádku
 10: špatný formát obrázku
 2x: chyba nastala při hledání vagonu
 22x: chyba nastala při lokalizaci textu
 3x: chyba nastala při čtení textu

Funkce *hledej_vagon(...)*

Funkce *hledej_vagon(...)* slouží pro nalezení hrany bez nutnosti volání celého čtení. Je samozřejmě volány i funkcí *cteni(...)*. Důvodem vytvoření *hledej_vagon(...)* jako samostatné funkce je možnost testu přítomnosti okraje vagónu a jeho polohy. Funkce *hledej_vagon(...)* vychází ze druhého způsobu lokalizace textu (kapitola 4.2.2.2), kdy jsou zjištěny souřadnice levého horního rohu vagónu.

Celá hlavička funkce:

int hledej_vagon (Image *im_in, Image *im_out, Point * hrana, int info, int *xhrany);

im_in – vstupní barevný obrázek

im_out – výstupní upravený obrázek

roh – bod na levém horním rohu vagónu

info – informace pro zpracovávání *im_out*

0 – nic

1 – kreslení čar výsledku

2 – kreslení čar LP

hrana – x souřadnice hrany (bez korekce)

chybové hodnoty: 1 – chyba LP; 2 – jiná chyba

5 MODUL ČTENÍ RZ NÁKLADNÍCH AUTOMOBILŮ

Druhým identifikačním modulem systému pro automatickou identifikaci je modul pro identifikaci registračních značek nákladních automobilů.

5.1 UMÍSTĚNÍ KAMERY

Původní umístění kamery na spojnici mezi rampami se ukázalo jako ne úplně vhodné. Kamera byla umístěna na odbavovací konstrukci, ve výšce přibližně pěti metrů. Její pohled byl orientován směrem k přijíždějícím nákladním vozům. Na obrázku 5.1 je vlevo vidět záběr z kamery a vpravo je zobrazená rampa, na které byla kamera umístěna.



obr. 5.1 – původní umístění kamery (vlevo záběr kamery, vpravo rampa)

Tato pozice byla ideální z hlediska lokalizace RZ a jejího čtení, protože značka nebyla téměř obrazově deformovaná. Jediné předzpracování obrazu, které bylo potřeba provést, byla rotace celého snímku.

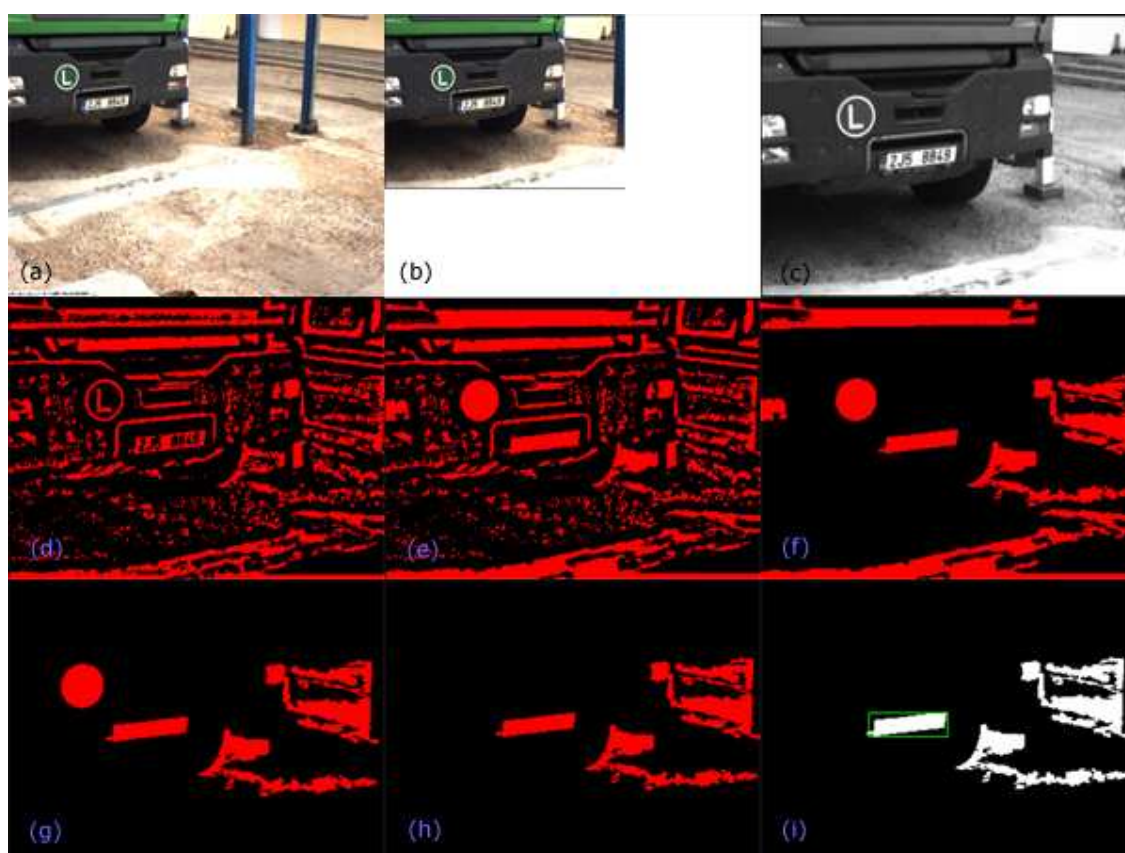
Snímek byl pořizován při příjezdu nákladního vozu, což se posléze ukázalo jako neřešitelný problém. V okamžiku, kdy byl přítomen jeden vůz na rampě a přijížděl další, bránil přívěs prvního vozu v pořízení snímku druhého, přijíždějícího nákladního automobilu. Řešením celé situace by bylo zaškolování řidičů o způsobu najíždění k rampě, ale toto by bylo vzhledem k množství řidičů, jejich časté obměně a příslušnosti k dodavatelským firmám nerealizovatelné. Přikročilo se tedy k přemístění kamery.

Při druhém a finálním umístění byla kamera připevněna na přední betonovou zábranu (přední z pohledu na obr. 5.1 vpravo). Tímto umístěním ale vzniklo mnoho problémů. Největším a nejzásadnějším je velké obrazové zkreslení vozu, jak je vidět na obrázku 5.2.



obr. 5.2 – záběr kamery na čtení RZ (finální umístění)

Již na první pohled je patrné, že lokalizace registrační značky je veliký problém. Tato značka při různém umístění na čele vozu zásadním způsobem mění své parametry, jako jsou rozměry, úhel zkosení a dokonce velikost obrazové plochy. Řešením této obtížné situace by bylo provedení kalibrace obrazu. Sehnání kalibrační desky o rozměrech čela nákladního automobilu by byl však velice obtížný úkol, ne-li neřešitelný. Tato kalibrace by také musela být provedena pro každou možnou pozici pořízení obrazu. Proto se ke kalibrování celého obrazu neprovádí. Kalibrace je nicméně použita a provádí se až pro lokalizovanou RZ.



obr. 5.3 – postup lokalizace RZ

5.2 LOKALIZACE REGISTRAČNÍ ZNAČKY

Lokalizace RZ je zásadní problém. Jak již bylo naznačeno, lokalizace RZ není možné provést pomocí hledání přesných tvarů nebo objektů v obraze. Proto je použita posloupnost binárních morfologických operací, pomocí které je provedena eliminace největšího množství objektů v obraze. Snahou je, aby jich na snímku zůstalo co možná nejmenší množství a aby parametry odpovídaly možným parametrům RZ. Celý postup je naznačen v následujícím textu. Přesný postup použití jednotlivých funkcí a jejich nastavení při lokalizaci RZ byl vytvořen pomocí aplikace Vision Assistant 8.0. Poté byla tato získaná posloupnost funkcí naprogramována v LabWindows/CVI.

První operací je provedeno omezení oblasti vyhledávání na prostor (obr. 5.3b), kde se může RZ nalézat. Tento prostor je zvolen v levé horní části snímku, kde je nejmenší obrazová deformace. Je také vymezena prostorem, ve kterém nadřazený systém lokalizuje kamion. Dalším krokem je získání šedotónového obrazu (obr. 5.3c) z původního barevného snímku. Je použit klasický převod (popis v kapitole 1.3.2) na Y model.

Na šedotónový obraz je následně použita funkce pro inteligentní prahování (obr. 5.3d) *imaqLocalTreshold(...)*, provádějící výpočet prahování v rámci malého okolí každého bodu (okolí velikosti 32x32 pixelů).

Na nově vzniklý binární obraz je již možné aplikovat složitější binární morfologické operace. Jednotlivé operace jsou obsaženy v knihovně NI-VISION, podsložka binární morfologické operace. První operací je vyloučení okrajových objektů z obrazu funkcí *imaqRejectBorder(...)* a poté vyplnění děr v objektech (obr. 5.3e) pomocí funkce *imaqFillHoles(...)*.

Takto předpřipravený obraz je poté zbaven malých objektů funkcí *imaqSizeFilter(...)* a dilatován funkcí *imaqMorphology(...)*. Výsledek je vidět na obr. 5.3f. Po opětovném provedení odstranění okrajových objektů (obr. 5.3g) jsou odstraněny i velké objekty (obr. 5.3h).

Nyní je obraz ve stavu, kdy je v nejlepším případě zobrazen jen objekt, který je obrazem RZ a odpovídá její poloze. Může samozřejmě nastat případ, že se v obraze stále nachází více objektů. K jejich nalezení se používá funkce *imaqMatchShape(...)*. Pro hledání je použita univerzální předloha RZ vzniklá upraveného snímku vozu s RZ umístěnou ve středu masky. Seznam nalezených objektů je předán jako pole struktury *ShapeReport*.

Struktura nese informace o rozměrech, pozici, středu a klasifikačního skóre nalezeného objektu.

Vhodným posuzováním jednotlivých vlastností nalezených objektů lze vyseparovat objekty, které jasně nemohou být považovány za obraz RZ. Vyhovující objekty postupují do konečné fáze, je na nich proveden pokus o čtení. V ideálním případě projde selekcí pouze obraz RZ. Pokud ne, začnou pokusy čtení s objektem s největším identifikačním skóre.

5.3 TRANSFORMACE RZ

Z původního šedotónového snímku je získán výřez z pozice nalezeného obrazu RZ. Pro získání lepších výsledků je provedeno čtyřnásobné zvětšení tohoto výběru (na obrázku 5.4 vlevo). Po pořízení binárního obrazu zvětšené RZ jsou získány úhly udávající zkosení značky, ze kterých je možné vytvořit kalibrační masku (na obrázku 5.4 vpravo).



obr. 5.4 – zvětšená RZ a kalibrační maska

Pro transformaci obrazové předlohy RZ je nejprve nutné získat potřebné informace z kalibrační masky pomocí funkce *imaqLearnCalibrationGrid(...)*. Tyto informace jsou uloženy přímo do obrazu s kalibrační maskou. Pomocí funkce *imaqCopyCalibrationInfo(...)* jsou přeneseny na transformovaný snímek a funkcí *imaqCorrectCalibratedImage(...)* je provedena kalibrace jako taková.



obr. 5.5 – nedeformovaná RZ a její předloha pro čtení

Po aplikaci kalibrace je získána zvětšená a nedeformovaná RZ (obr. 5.5 vlevo). Před samotným přečtením textu RZ je nutné oříznutí přebytečné části obrazu (obr. 5.5 vlevo) tak,

aby zůstala v obraze jen RZ. Po ořezu je provedeno lokální prahování metodou NIBlack (obr. 5.5 vpravo).

5.4 ČTENÍ RZ

Finální fází identifikace nákladních automobilů je přečtení lokalizované a předzpracované registrační značky. Znaková sada pro čtení RZ byla vytvořena pomocí aplikace NI OCR Training Interface popsané v kapitole 3.2.2. Abeceda je kombinovaná jak pro identifikace českých RZ, tak německých. Také znaková sada neobsahuje všechna písmena abecedy, obsahuje pouze ta, která se v průběhu vyhodnocování objevila a byla identifikovatelná.

Po přečtení je nadřazenému systému vrácen přečtený řetězec a pole hodnot reprezentující identifikační skóre jednotlivých rozpoznaných znaků. Pokud se čtení nepodaří nebo nejsou hodnoty uspokojivé, je přikročeno k opakování celého postupu kalibrace a čtení s dalším objektem nalezeným při lokalizaci RZ. Pokud ale již není žádný další nalezený objekt, je identifikace ukončena bez rozpoznaného textu.

5.5 POPIS FUNKCE NAJDİRZ(...)

Modul identifikace registračních značek nákladních automobilů je obsluhován nadřazeným systémem pomocí jediné funkce, *najdiRZ(...)*. Celý modul je obsažen ve zdrojovém souboru *autaOCR.c* spolu s hlavičkovým souborem *autaOCR.h*, který obsahuje i definice jednotlivých konstant, použitých při identifikaci. Této funkci je předán obraz pořízený kamerou *cam1*. Nadřazenému systému je předána rozpoznaná RZ nákladního automobilu, identifikační skóre pro každý rozpoznaný znak a snímek RZ.

Úplná hlavička funkce:

*int najdiRZ (Image *im_in, Image *znacka, char *textRZ, int *prav)*

im_in – vstupní barevný obrázek
znacka – výstupní šedý obrázek RZ
textRZ – textový řetězec přečtené RZ
prav – pole obsahující hodnoty identifikačního skóre

navratove hodnoty:

- 0 – identifikováno
- 1 – neporadilo se nalézt RZ
- 2 – nalezena RZ není dost široká
- 3 – nalezena RZ není dost vysoká
- 4 – rezervováno (chyba kalibrace)
- 5 – chyba výběru po kalibraci

6 ZÁVĚR

V rámci diplomové práce vznikly dva samostatné moduly pro čtení vagónů a RZ, z nichž každý zajišťuje lokalizaci a přečtení textu v obraze. Oba vycházejí z přesného umístění kamery a při identifikaci využívají určité neměnné parametry, odpovídající obrazovým předlohám statických objektů na pořízených snímcích. Pro potenciální přednastavení vyhledávacích parametrů je možnost jejich změny v příslušných hlavičkových souborech modulů bez nutnosti úprav celého programu.

Celý systém byl postupně nasazován do praxe od března 2007, kdy byly získány první záběry z kamer a docházelo k základním pokusům identifikace vagónů. V této době nebyly výsledky nijak oslňující a i přes velké omezení identifikovatelných typů vagónů pouze na malý a velký uhlák ČD, se úspěšnost čtení pohybovala okolo 40 %. Postupnými úpravami se rozšiřovalo spektrum identifikovatelných vagónů a pravděpodobnost jejich rozpoznání se zvyšovala. Po překonání několika zásadních problémů a odladění mnoha nedokonalostí se podařilo dostat na současných 80 % rozpoznaných identifikovatelných vagónů. Takové výsledky program poskytuje, pokud celý systém běží stabilně a za příznivých podmínek.

Pokud jsou splněny všechny předpoklady pro úspěšnou identifikaci, je text na snímku nalezen a přečten ve většině případů. Tento rozdíl v úspěšnosti je zapříčiněn především shodou několika faktorů. Patří mezi ně hardwarové i technické problémy a např. mezní světelné podmínky. Mezi hardwarové problémy patří výpadky komunikace s kamerami a nestabilita sítě (výpadky elektrické energie). Mezi největší technické problémy patří rosa na odrazce pro optické závory. Když je odrazka zamlžená, nefungují optické závory a není tedy možné rozpoznat přítomný vlak. Dalším problémem jsou nečistoty na sklech krytů kamer. Kvůli těmto nečistotám dochází ke snížení kontrastu i jasů pořízených snímků. Dalším problémem kamer samotných je jejich přehřívání při vystavení dlouhodobého působení přímého slunce na kryt kamery. Při přehřátí dochází k chybnému rozlišování barev, což zapříčiní výpadek zelených světlocitlivých prvků. Na výsledném obraze je tato porucha jasně viditelná (příklad v příloze 1). Dalším problémem jsou mezní světelné podmínky, kdy různé pozice slunce (svítání, přímé polední slunce, soumrak, protisvětlo) významně mění jas a kontrast na snímku a jeho vyhodnocení se stává obtížnější. Díky rozsáhlosti programu bylo nutné řešit celou řadu problémů, ale s většinou z nich jsem se vypořádal a eliminoval je. Některé však nelze plně vyloučit.

Ne vždy se ale podaří vagón identifikovat. Je ještě jeden aspekt pořízení obrazu, který v této práci nebyl zmíněn a zároveň má zásadní význam pro celé zpracování a vyhodnocení. Je to barevná kvalita snímaného objektu. Pokud je text na vagónu nečitelný nebo poškozený, sebelepší automatický systém si s ním nemůže poradit. Některé z takto nekvalitních textů jsou uvedeny na obrázcích v příloze č. 1.

Na druhou stranu je patrná síla tohoto automatického systému v případech, kdy je obraz pořízen například za špatných světelných podmínek a text je pouhým okem nečitelný. Přesto bývá takovýto vůz správně identifikován.

Druhou částí celé práce je automatická identifikace registračních značek nákladních automobilů. Stejné problémy jako při identifikaci vagónů nastávají i při identifikaci registračních značek nákladních automobilů. Zde je tento problém ještě umocněn jistě „náhodami“, kdy RZ jsou přidělány pomocí zrezavělých šroubů v místech, kde mění či zasahují do textu. Dále je ještě problém s natočením RZ, čímž dochází ke výraznému znesnadnění její identifikace.

Její spolehlivá identifikace je zajištěna pomocí panelu, který je zobrazen operátorovi při zjištění přítomnosti nákladního vozu. Obsahuje informace o rozpoznané RZ. Je to přečtený text, její obraz a zároveň je zobrazen i celý pořízený snímek, pro případ špatné lokalizace RZ. Poté operátor potvrdí nebo opraví její text.

V budoucnu je možné celý systém stále zdokonalovat a zlepšovat jeho schopnosti. Pomocí systému lokalizace textu na vagónech hledáním masek, je v budoucnu možné i rozlišit dosud neidentifikovatelné typy vagónů a použít na jejich čtení díky tomuto rozlišování originální znakové sady. Tím je možné zajistit automatickou identifikaci téměř všech typů vozů. Pokud se podaří udržet současné dobré a budoucí jistě lepší výsledky, vyvstává zde možnost vytvoření obdobných systémů i pro jiné pobočky firmy NEZÁVISLÁ PŘEJÍMKA s.r.o..

POUŽITÉ ZDROJE

- [1] HLAVÁČ Václav, SEDLÁČEK Miloš: Zpracování signálů a obrazů, Vydavatelství ČVUT, Praha 2005, ISBN 80-01-03110-1
- [2] ŠONKA Milan, HLAVÁČ Václav, BOYLE Roger: Image Processing, Analysis, and Machine Vision, Third Edition, Thomson Learning, 2008, ISBN-10: 0-495-24438-4
- [3] KOTEK, Z., MAŘÍK, V., HLAVÁČ, V., PSUTKA, J., ZDRÁHAL, Z., Metody rozpoznávání a jejich aplikace, ACADEMIA, Praha 1993, ISBN 80-200-0297-9
- [4] BRUCE Eckel, Myslíme v jazyku C++, GRADA Publishing, Praha 2000, ISBN 80-247-9009-2
- [5] VIRIUS Miroslav, Pasti a propasti jazyka C++, CP Books, Brno 2005, ISBN 80-251-0509-1
- [6] NAVAJO otevřená encyklopedie, Viditelné spektrum [online], [citováno 1.5.2008], dostupné z <http://viditelne-spektrum.navajo.cz/>
- [7] PIHAN Roman. Barva a vyvážení bílé - 1. Barva a její obraz v PC [online], 29.06.2006, [citováno 1.5.2008], dostupný z http://www.fotografovani.cz/art/fozak_df/rom_color1.html
- [8] FIŘT Jaroslav, HOLOTA Radek, Digitalizace a zpracování obrazu [online], [citováno 1.5.2008], dostupné z <http://home.zcu.cz/~holota5/publ/DigZprO.pdf>
- [9] Wikipedia otevřená encyklopedie, Bilineární transformace [online], [citováno 1.5.2008], dostupné z http://cs.wikipedia.org/wiki/Bilineární_interpolace
- [10] ČEPELÁK Jiří, Řešené příklady v jazyku C, Computer Press, Praha 2001, ISBN 80-7226-575-X
- [11] HARBISON S. P., STEELE G. L. Jr., Referenční příručka jazyka C, SCIENCE, Veletiny 1996, ISBN 80-901475-50
- [12] Wikipedia otevřená encyklopedie, CCD [online], [citováno 1.5.2008], dostupné z <http://cs.wikipedia.org/wiki/CCD>
- [13] Wikipedia otevřená encyklopedie, Power on Ethernet [online], [citováno 1.5.2008], dostupné z http://cs.wikipedia.org/wiki/Power_over_Ethernet
- [14] Wikipedia otevřená encyklopedie, Bayerův barevný filtr, [citováno 1.5.2008], dostupné z http://en.wikipedia.org/wiki/Bayer_filter

Seznam obrázků:

obr. 1.1 – Vzorkovací mřížky	- 11 -
obr. 1.2 – Optická škála	- 12 -
obr. 1.3 – Citlivost lidského oka na barvy	- 13 -
obr. 1.3 – Prostor RGB	- 13 -
obr. 1.4 – Rozdělení jasů v Y modelu	- 14 -
obr. 1.5 – Model HSL	- 15 -
obr. 1.6 – Základní šedotónové transformace	- 17 -
obr. 1.7 – Odprahovaný obraz (7% práh)	- 18 -
obr. 1.8 – Použití metody NIBlack	- 19 -
obr. 1.9 – Interpolace metodou nejbližšího souseda	- 22 -
obr. 1.10 – Lineární interpolace	- 23 -
obr. 1.11 – Bilineární interpolace	- 24 -
obr. 1.12 – Jádru bikubické interpolace	- 24 -
obr. 1.13 – Elementy pro binární matematickou morfologii	- 25 -
obr. 1.14 – Eroze – (a) vstupní obraz, (b) strukturní element, (c) eroze	- 25 -
obr. 1.15 – Dilatace – (a) vstupní obraz, (b) strukturní element, (c) dilatace	- 26 -
obr. 1.16 – Otevření – (a) vstupní obraz, (b) strukturní element, (c) otevření	- 26 -
obr. 1.17 – Uzavření – (a) vstupní obraz, (b) strukturní element, (c) uzavření	- 26 -
obr. 2.1 – IEEE 1394 – konektor, opakovač BELKIN	- 28 -
obr. 2.2 – Kabel a koncovka UTP 5e	- 29 -
obr. 2.3 – Kamera Guppy F-080C	- 29 -
obr. 2.4 – Kamera Basler Scout csA1000-30gc	- 30 -
obr. 2.5 – Záběry použitých kamer	- 33 -
obr. 2.6 – Reflexní závory – instalace	- 34 -
obr. 2.7 – Optosenzor QS30LV	- 34 -
obr. 2.8 – PC I/O karta NI 6519	- 35 -
obr. 2.9 – Schématický plán kamerového systému	- 36 -
obr. 3.1 – Základní okno aplikace NI OCR Training Interface	- 41 -
obr. 3.2 – Příklady nastavení prahovacího módu	- 42 -
obr. 3.3 – Záložka Threshold	- 43 -
obr. 3.4 – Záložka Size & Spacing	- 43 -
obr. 3.5 – Záložka Read Options	- 44 -
obr. 3.6 – Tabulka výsledků čtení Results	- 44 -
obr. 3.7 – Způsob učení znaků	- 45 -
obr. 3.8 – Záložka editace abecedy	- 45 -
obr. 3.9 – Okno aplikace Vision Assistant 8.0	- 47 -
obr. 3.10 – Záložky pro přehlednost funkcí	- 47 -
obr. 4.1 – Různé fonty vagónů	- 49 -
obr. 4.2 – Masky pro vyhledání textu	- 52 -
obr. 4.3 – Nalezený text a označený výběr pro čtení	- 53 -
obr. 4.4 – Označení pozic jednotlivých jasových profilů a nalezená hrana	- 54 -
obr. 4.5 – Hledání hrany vagónu	- 55 -
obr. 4.6 – Nalezení výšky vagónu	- 56 -
obr. 4.7 – Hledání horního lemu vagónu	- 57 -
obr. 4.8 – Práhovaný obraz (10% nejjasnějších bodů)	- 58 -
obr. 4.9 – Průběh součtu podle osy x (šířka textu)	- 58 -
obr. 4.10 – Průběh součtu podle osy y	- 59 -
obr. 4.11 – Třetí řádek textu	- 60 -
obr. 5.1 – Původní umístění kamery	- 62 -
obr. 5.2 – Záběr kamery na čtení RZ (finální umístění)	- 63 -
obr. 5.3 – Postup lokalizace RZ	- 63 -
obr. 5.4 – Zvětšená RZ a kalibrační maska	- 65 -
obr. 5.5 – Nedeformovaná RZ a její předloha pro čtení	- 65 -

příloha 1 – galerie nákladních vagónů

Vozy ČD



Velký uhlák (červený)



Malý uhlák



Velký uhlák (modrý)



Velký uhlák (modrý, typ 2)



Klaničák ČD



Velký uhlák (typ 3)

Vozy Polských drah a ŽSR



Velký uhlák Polských drah



Velký uhlák ŽSR

Vozy Deutsche Bahn



Špatně identifikovatelné texty



Chyba barev v důsledku přehřátí kamery



snímek z přehřáté kamery

detail snímku

příloha 2 – příklad snímků RZ



příklady špatně identifikovatelných nebo poškozených registračních značek



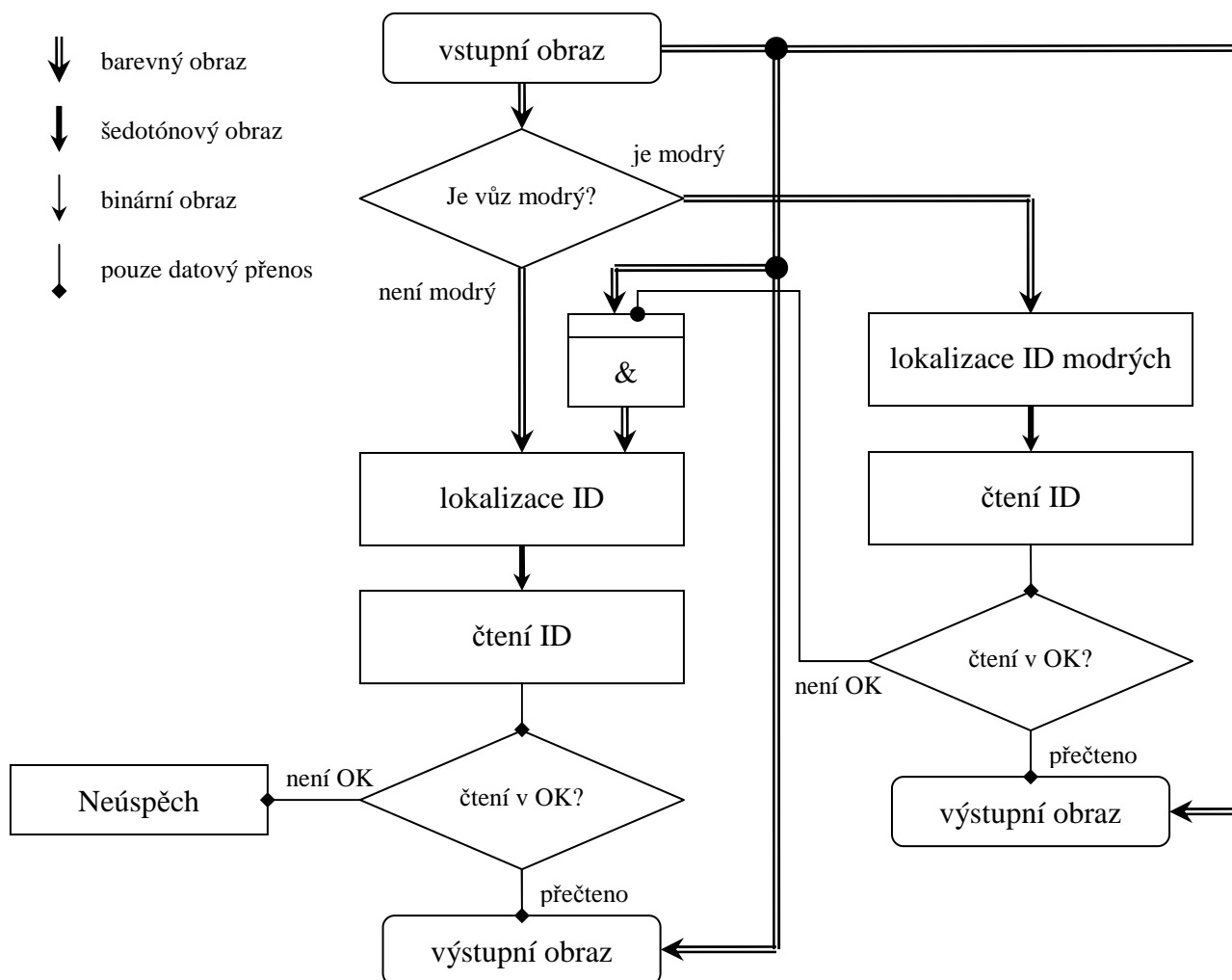
příklady nakloněných RZ – velké optické zkreslení



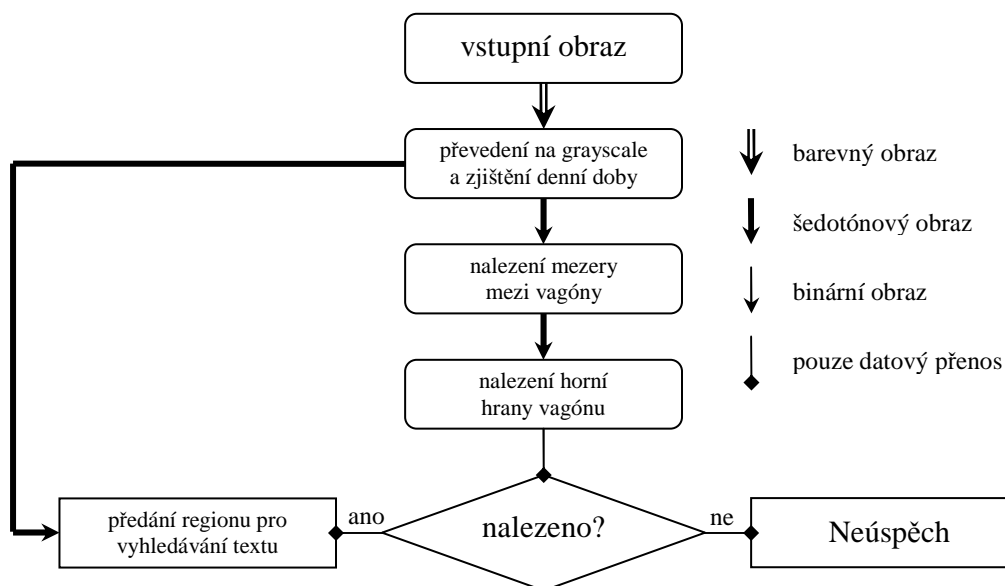
příklady špatných světelných podmínek + nelokalizovatelná RZ (2P1 5957)

příloha 3 – diagramy jednotlivých modulů

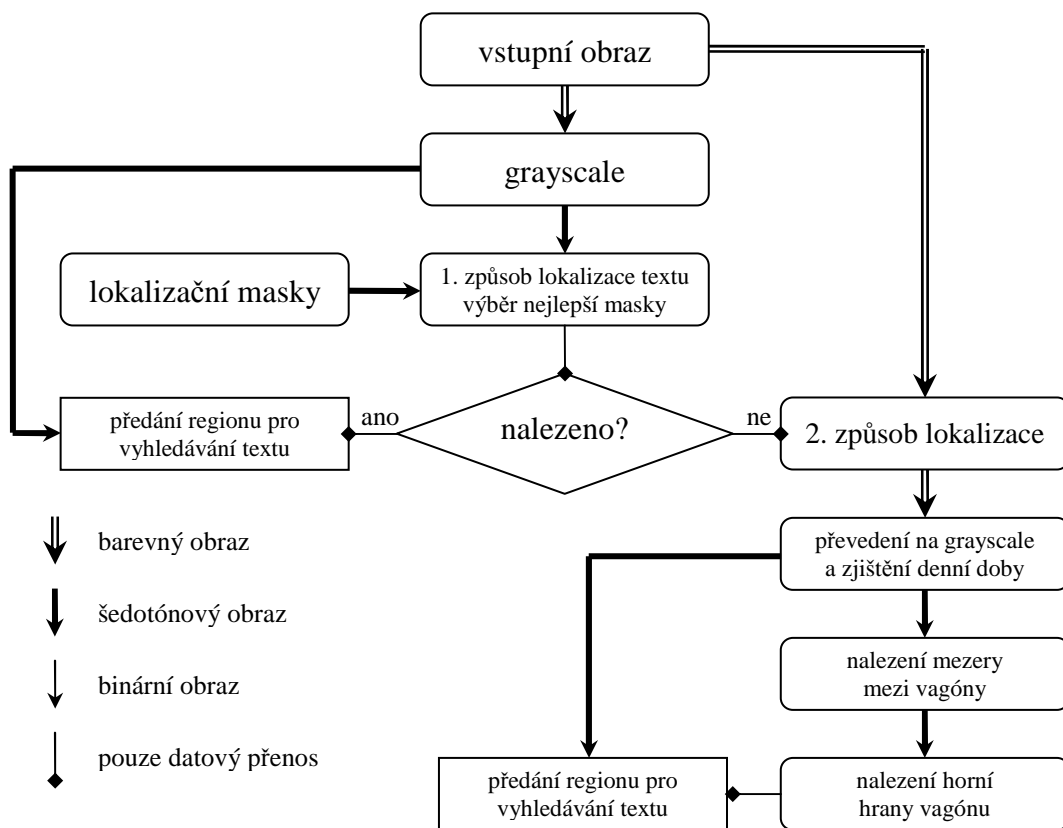
Diagram algoritmu identifikace vagónu



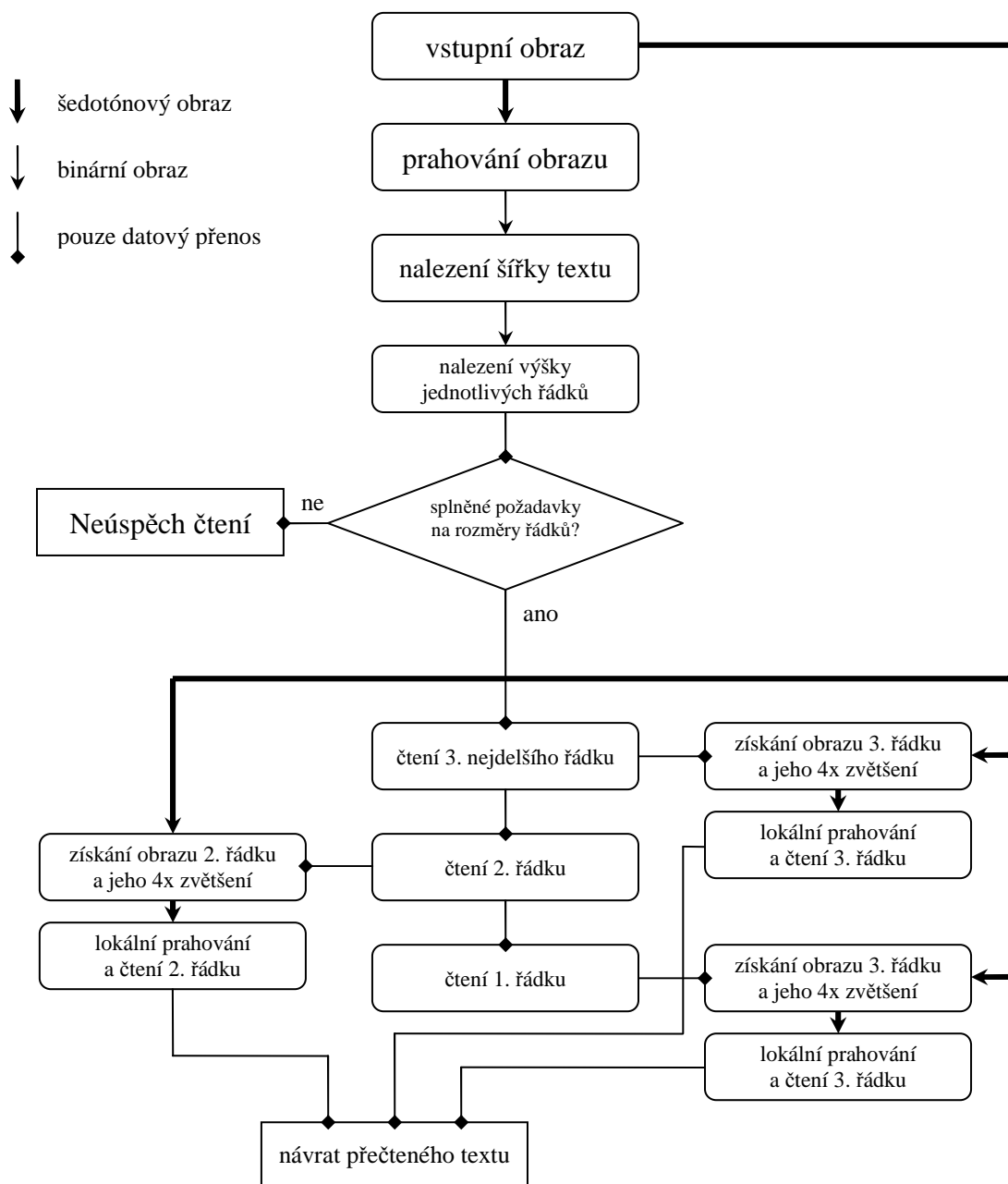
Lokalizace ID modrých uhláků



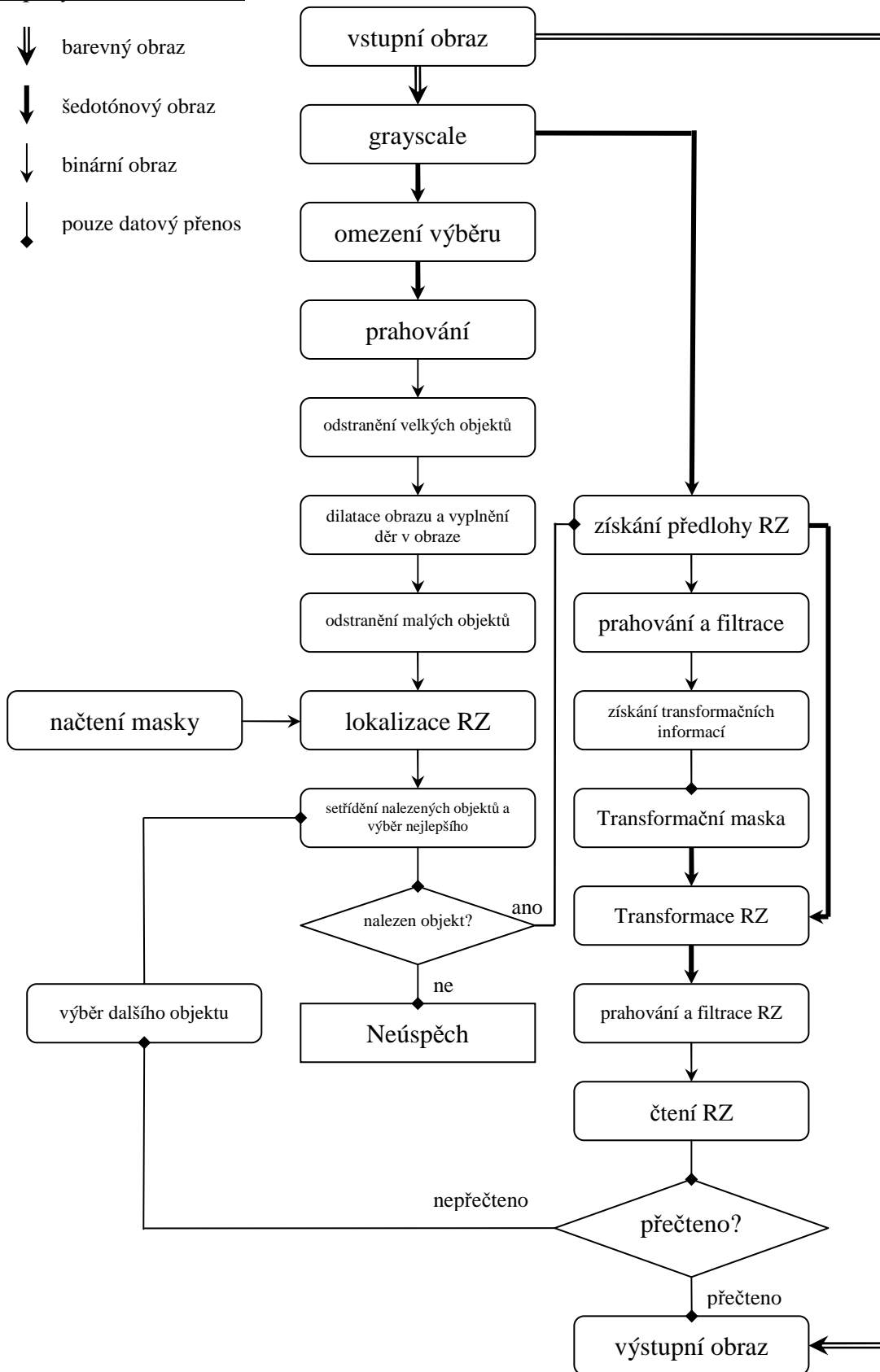
Lokalizace ID



čtení ID



Postup vyhodnocení RZ



příloha 4 – popis použitých funkcí

[fce 01]

- funkce pro filtraci vstupního pole Butterworthova dolnoproustního digitálního filtru

AnalysisLibErrType **Bw_LPF** (const double Input_Array[], int Number_of_Elements, double Sampling_Frequency, double Cutoff_Frequency, int Order, double Output_Array[]);

Input_Array – vstupní hodnoty filtrace

Number_of_Elements – počet prvků v poli

Sampling_Frequency – vzorkovací frekvence

Cutoff_Frequency – mezní frekvence

Order – řád filtru

Output_Array – výstupní hodnoty po filtraci

[fce 02]

- funkce pro aplikaci kalibračních informací

int **imaqCopyCalibrationInfo**(Image* dest, const Image* source);

dest – pointer na obrázek s kalibračními informacemi

source – pointer na obrázek pro kalibraci

[fce 03]

- funkce pro aplikaci kalibrace

int **imaqCorrectCalibratedImage**(Image* dest, const Image* source, PixelValue fill, InterpolationMethod method, const ROI* roi);

dest – pointer na obrázek s kalibračními informacemi

source – pointer na obrázek pro kalibraci

fill – určení hodnoty prázdných pixelů

method – interpolační metoda

roi – oblast zájmů (možnost omezení výběru)

[fce 04]

- funkce pro vytvoření znakové sady

CharSet* **imaqCreateCharSet**();

[fce 05]

- funkce pro vyplnění děr v binárním obraze

int **imaqFillHoles**(Image* dest, const Image* source, int connectivity8);

dest – pointer na cílový obrázek

source – pointer na zdrojový obrázek

connectivity8 – nastavení propojitelnosti pixelu s okolím 4, 8 okolí

[fce 06]

- funkce pro vyplnění děr v binárním obraze

PatternMatch* **imaqFindPattern**(Image* image, Image* pattern, RotatedRect searchRect, const FindPatternOptions* options, const CoordinateTransform2* transform, int* numMatches);

image – pointer na obrázek, ve kterém je vyhledáváno

pattern – vyhledávaná předloha (nutnost předzpracování fcí *imaqLearnPattern*)

searchRect – region, ve kterém je vyhledáváno

options – definice způsobu vyhledávání

transform – definice transformace souřadných systémů mezi obrazem a předlohou

numMatches – počet vyhledávaných objektů

[fce 07]

- funkce pro změnu barevného prostoru

Color2 **imaqChangeColorSpace2**(const Color2* sourceColor, ColorMode sourceSpace, ColorMode destSpace, double offset, const CIEXYZValue* whiteReference);;

sourceColor – pointer na zdrojový pixel
sourceSpace – pointer na zdrojový barevný prostor
destSpace – pointer na cílový barevný prostor
offset – pro převod na HSL udává posun v barevnosti složky H
whiteReference – vyvážení bílé

[fce 08]

- funkce pro získání kalibračních informací z kalibračního snímku

```
int imaqLearnCalibrationGrid(Image* image, const ROI* roi, const LearnCalibrationOptions* options, const
    GridDescriptor* grid, const CoordinateSystem* system, const RangeFloat* range,
    float* quality);
```

image – pointer na kalibrační snímek
roi – oblast zájmů (možnost omezení výběru)
options – definice způsobu vyhledávání
grid – nastavení parametrů kalibrační mřížky
system – definice reálného systému souřadnic
range – nastavení min. a max. vzdálenosti kalibračních bodů
quality – udává skóre naučení kalibrace (0..1000)

[fce 09]

- funkce pro získání vyhledávacích informací z masky

```
int imaqLearnPattern(Image* image, LearningMode learningMode);
    image – pointer na předlohu pro lokalizaci (obrázek)
    learningMode – informace o možném posunu předlohy (posun, rotace, obě)
```

[fce 10]

- funkce pro získání jasového profilu na přímce

```
LineProfile* imaqLineProfile(const Image* image, Point start, Point end);
```

funkce pro získání jasových hodnot na přímce

image – pointer na obrázek, ve kterém je vyhledáváno
start – bod počátku profilové čáry
end – bod konce profilové čáry

[fce 11]

- funkce pro získání binárního obrazu lokálním prahováním

```
int imaqLocalThreshold(Image* dest, const Image* source, unsigned int windowWidth, unsigned int
    windowHeight, LocalThresholdMethod method, double deviationWeight, ObjectType
    type, float replaceValue);
```

dest – pointer na cílový obrázek
source – pointer na zdrojový obrázek
windowWidth – šířka okna pro lokální threshold
method – volba prahovací metody
windowHeight – výška okna pro lokální threshold
deviationWeight – hodnota konstanty k pro NIBlack
ObjectType – určení hledání světlých nebo tmavých objektů
replaceValue – nahrazovací hodnota

[fce 11]

- funkce pro hledání objektů v obraze

```
ShapeReport* imaqMatchShape(Image* dest, Image* source, const Image* templateImage, int scaleInvariant,
    int connectivity8, double tolerance, int* numMatches);
```

dest – pointer na cílový obrázek
source – pointer na zdrojový obrázek
templateImage – obraz hledané předlohy
scaleInvariant – hledání bez ohledu na velikost
connectivity8 – nastavení propojitelnosti pixelu s okolím 4, 8 okolí

tolerance – udává stupeň shody nalezeného a hledaného objektu
numMatches – počet hledaných objektů

[fce 12]

- funkce pro binární matematickou morfologii

int **imaqMorphology**(Image* dest, Image* source, MorphologyMethod method, const StructuringElement* structuringElement);

dest – pointer na cílový obrázek
source – pointer na zdrojový obrázek
method – výběr morfologické metody
structuringElement – strukturní element

[fce 12]

- funkce pro načtení souboru s abecedou

int **imaqReadOCRFile**(const char* fileName, CharSet* set, String255 setDescription, ReadTextOptions* readOptions, OCRProcessingOptions* processingOptions, OCRSpacingOptions* spacingOptions);

fileName – pointer na adresu abecedy
set – pointer na abecedu
setDescription – vrací informační text z abecedy
readOptions – nastavení pro čtení
processingOptions – nastavení pro zpracování
spacingOptions – nastavení pro rozměrů znaků

[fce 13]

- funkce pro čtení textu

ReadTextReport2* **imaqReadText2**(const Image* image, const CharSet* set, const ROI* roi, const ReadTextOptions* readOptions, const OCRProcessingOptions* processingOptions, const OCRSpacingOptions* spacingOptions);

image – pointer na obrázek, ve kterém je vyhledáváno
set – pointer na abecedu
roi – oblast zájmu
readOptions – nastavení pro čtení
processingOptions – nastavení pro zpracování
spacingOptions – nastavení pro rozměrů znaků

[fce 14]

- funkce pro odstranění okrajových objektů

int **imaqRejectBorder**(Image* dest, Image* source, int connectivity8);

image – pointer na obrázek
connectivity8 – nastavení propojitelnosti pixelu s okolím 4, 8 okolí

[fce 15]

- funkce pro změnu rozměrů obrazu

int **imaqResample**(Image* dest, const Image* source, int newWidth, int newHeight, InterpolationMethod method, Rect rect);

dest – pointer na cílový obrázek
source – pointer na zdrojový obrázek
newWidth – nová šířka
newHeight – nová výška
method – použitá metoda jasové aproximace
rect – obdélník pro změnu

[fce 16]

- funkce pro odstranění malých nebo velkých objektů

```
int imaqSizeFilter(Image* dest, Image* source, int connectivity8, int erosions, SizeType keepSize, const StructuringElement* structuringElement);
```

dest – pointer na cílový obrázek

source – pointer na zdrojový obrázek

connectivity8 – nastavení propojitelnosti pixelu s okolím 4, 8 okolí

erosions – počet opakovaných erozí

keepSize – odstranění malých nebo velkých objektů

structuringElement – strukturní element

[fce 17]

- funkce pro odstranění malých nebo velkých objektů

```
int imaqTrainChars(const Image* image, CharSet* set, int index, const char* charValue, const ROI* roi, const OCRProcessingOptions* processingOptions, const OCRSpacingOptions* spacingOptions);
```

image – pointer na obrázek, ve kterém je vyhledáváno

set – pointer na abecedu

charValue – význam znaku

roi – oblast zájmu

processingOptions – nastavení pro zpracování

spacingOptions – nastavení pro rozměrů znaků

[fce 18]

- funkce pro zápis znakové abecedy

```
int imaqWriteOCRFile(const char* fileName, const CharSet* set, const char* setDescription, const ReadTextOptions* readOptions, const OCRProcessingOptions* processingOptions, const OCRSpacingOptions* spacingOptions);
```

fileName – pointer na adresu abecedy

set – pointer na abecedu

setDescription – vrací informační text z abecedy

readOptions – nastavení pro čtení

processingOptions – nastavení pro zpracování

spacingOptions – nastavení pro rozměrů znaků

Podrobnější informace o funkcích na přiloženém CD nebo na nationalinstruments.com